DOCUMENT RESUME

ED 190 645 TM 800 441

AUTHOR Veldman, Donald J.
TITLE The PRIME System: Computer Programs for Statistical Analysis.
INSTITUTION Texas Univ., Austin. Research and Development Center for Teacher Education.
SPONS AGENCY National Inst. of Education (DHEW), Washington, D.C.
PUB DATE Sep 78
NOTE 106p.

EDRS PRICE MF01/PC05 Plus Postage.
DESCRIPTORS Analysis of Variance; Cluster Analysis; *Computer Programs; Correlation; Equated Scores; Factor Analysis; Input Output; Item Analysis; Multiple Regression Analysis; Scoring; *Statistical Analysis; Weighted Scores
IDENTIFIERS Chi Square; PRIME System

ABSTRACT

PRIME is a library of 44 batch-oriented computer routines: 20 major package programs, which use 12 statistical utility routines, and 12 other utility routines for input/output and data manipulation. This manual contains a general description of data preparation and coding, standard control cards, input deck arrangement, standard options, and reporting statistics. All 44 routines are then described. The major programs are written as subroutines without arguments, but must be summoned by a simple FORTRAN driver program. The major programs perform: item analysis, Likert scaling, Rasch Model calibration; descriptive statistics and Chi-squared analysis; transformations and composites; cluster analysis; analysis of variance; intercorrelation analysis; regression analysis; and factor analysis. (CP)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED190645

U S DEPARTMENT OF HEALTH
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

C. Walton

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

The PRIME System

Computer Programs for Statistical Analysis

Donald J. Veldman

Correlates of Effective Teaching Program

The Research and Development Center for Teacher Education

The University of Texas

Austin, Texas

September 1978

Preparation of this manual was supported by a grant from the National Institute of Education, Department of Health, Education, and Welfare. However, the opinions expressed herein do not necessarily reflect the position or policy of the National Institute of Education, and no official endorsement by the National Institute of Education should be inferred.

MAR 1 4 1980

Table of Contents

	Page Number
System Design and Features	1
Data Preparation and Coding	2
Standard Control Cards	5
Input Deck Arrangement	6
Standard Options	10
Reporting Statistics	11
Data Modification and Generation	12
Test Scoring and Item Analysis	15
SCALES: Item analysis and Likert scaling	15
RASCH: Rasch model calibration and scoring	18
Descriptive Statistics and Chi-Square	21
UNITAB: Basic statistics and frequency distributions	21
DUOTAB: Frequency cross-tabulation and chi-square	23
CHICHI: Chi-square analysis of frequency data	25
Transformations and Composites	26
REFORM: Standard score transformations	27
SCORES: Weighted composite scoring	29
Cluster Analysis	32
GROUPS: Hierarchical grouping analysis	33
ASSIGN: Euclidean distance scoring	35
Analysis of Variance	38
AOV123: Univariate analysis of variance	39
MANOVA: Multivariate analysis of variance	42
INTRAR: Intraclass correlation analysis	45

Intercorrelation Analysis 47

MISCOR: Intercorrelation with pair-wise deletion 49

SELCOR: Cross-correlation with pair-wise deletion 50

CANONR: Canonical correlation analysis 52

Regression Analysis 55

REGRAN: Multiple correlation analysis 59

COVARY: Analysis of covariance 63

COMMAP: Commonality analysis 69

Factor Analysis 73

FACTOR: Principal components and image analysis 75

RELATE: Factor structure comparison 79

Statistical Utility Routines 81

SUMX: Sum or sum of squares of a vector 82

SCAP: Scalar product of two vectors 82

MNORM: Normalize matrix rows or columns 83

MPROD: Matrix multiplication 84

MINVS: Matrix inversion 85

ASCOR: Means, sigmas, intercorrelations 85

RTOGM: Convert R matrix to G 86

ROOTS: Roots and vectors of a symmetric matrix 87

ROOTA: Roots and vectors of an asymmetric matrix 88

VARRO: Varimax rotation 88

SOLVE: Iterative solution of a regression equation 89

PROB: Chance probability of an F ratio 90

Input/Output Utility Routines 90

READC: Read and print initial control cards 91

READM: Read and print option signal card 91

READT: Read and reflect a triangular matrix 92

READX: Read data for one subject 93

SPLIT: Break a five-digit integer into five one-digit integers 93

MDAT: Test a score for validity 94

ARRAY: Print and/or punch an array 94

SCORP: Punch a score vector for one subject 95

SWAP: Trade contents of two locations 96

ALTER: Dummy routine for UNITAB 96

MCARD: Read a model card for REGRAN 96

SORT: Sort a file of data using SRTMRG 97

Acquisition of Source Code 99

References 100

System Design and Features

The PRIME system is the most recent step in the evolution of a library of batch-oriented statistical programs that originated in 1961. The first published version of the system was in book form (Veldman, 1967). Since then, two extensive revisions -- EDSTAT-V (1971) and VSTAT (1974) -- have been installed locally and at a number of other academic institutions.

The R and D Center for Teacher Education has provided support for most of this development, as well as the release of this manual. The PRIME system is so named because Project PRIME (Kaufman, Agard, and Semmel, in press) provided the stimulus for the author to completely rewrite the entire library of programs and utility routines.

Users familiar with the older systems should be warned that some of the PRIME routines carry the same names as in earlier systems, but operate somewhat differently, particularly with regard to control cards and available options.

Organization of the System

PRIME consists of 20 major "package" programs, which employ 12 statistical utility routines and 12 other utility routines for input/output and data manipulation. All 44 routines are described in this manual. Most users will need only the information about the 20 major programs in order to use the system successfully, but programmers may find the utility routines useful in constructing special-purpose programs.

The PRIME system is stored on permanent file 7037 in the CDC 6600/6400 system at the University of Texas Computation Center. The major programs are written in the form of subroutines without arguments, but

have the status of independent programs. They must be called into operation by a simple Fortran "driver" program. The reason for structuring the system in this manner is to allow the user to override (replace) any of the standard utility routines with his own version if he wishes. One of these routines in particular is used in the system just so that it can be replaced under a variety of special circumstances. This will be explained in the section on Data Modification and Generation.

General Features of the System

PRIME is a batch-oriented system but has been used successfully through interactive terminals. It is basically card-oriented also, but is easily adapted to process data stored on tapes or permanent files. The programs are designed so that a series of independent problem decks can be "stacked" for a single run of the program.

Compared to other statistical program libraries, PRIME is distinguished by its ease of use, compact output, and consistency of options among programs. It is designed to include the techniques most widely used in behavioral sciences research, but researchers in other disciplines have found it to meet their needs as well.

Data Preparation and Coding

Although the programs in the PRIME system will accept data in any range and in any format, there are some conventions that will simplify use of the system if kept in mind when planning the collection and coding of research data.

Score Ranges

If possible, design instrumentation to yield one- or two-digit pos-

itive whole numbers. The metrics of most psychological measures are arbitrary, and seldom are they meaningful to more than two significant digits, so this suggestion is not really as restrictive as it sounds. Programs SCALES, UNITAB, and DUOTAB are the only ones in the system for which score ranges are relevant, and they can be adapted to handle scores of any kind, if necessary.

Coding Forms

If possible, design instrumentation so as not to require separate answer sheets, and so that punching can be done directly from the forms, so as to reduce potential errors. Avoid arbitrarily coded ranges; e.g. ask for number of years of schooling or income (in thousands) directly, to obtain the maximum unbiased information. One source of respondent errors is the use of random or systematic changes in the directionality of items in a series; such attempts to avoid response sets probably do more harm than good.

Missing Data

Do not use a special code such as "99" to indicate missing data, although the programs can be adapted to handle data prepared with such conventions. The programs that do permit automatic exclusion of missing data have three options: (1) all fields valid, (2) both zero and blank fields not valid, and (3) only blank fields not valid. It is best to adopt the same convention for all fields, e.g. a blank field means missing data in all cases.

Identification Fields

Card formats should include subject identification as the first field(s) on the card. Those programs which provide output of punched

cards for subjects allow transfer of up to eight consecutive columns of subject identification. Those programs are designed to expect such a field first in the format, i.e. before any score-field designations.

It is generally wise to reserve particular card columns to record such things as group membership, testing time of repeated measures, and even a code designating which of a set of tests the card represents. Such codes simplify card sorting necessitated for entry to some of the programs.

Card Formatting

Data card fields can be of any length, with any number of decimal places. To the extent possible without wasting a great deal of space, consecutive fields should be the same in length and number of decimal places in order to simplify designation of format to the programs.

Skipped columns between fields are not necessary, but they do make the cards or a listing of them easier to read. Decimal points need not be punched if placement in the field is consistent for all subjects. Plus signs are not necessary, but minus signs are, and require a separate column. Leading zeroes are not needed; leave such columns blank.

Data from each subject must occupy a separate card or series of cards. If repeated measures are obtained, begin the series of scores from each testing on a separate card, with the same format.

Tapes and Permanent Files

Although the programs are designed for data to be entered from cards, they can be adapted to access data from tape or disk with little difficulty. If the data set is relatively large, i.e. more than a box of cards, consideration should be given to the use of tape. The expense is

minimal, and errors of card reading and card wear are eliminated. Processing through interactive terminals is also made possible. The manner in which data are accessed from tape is explained in the section on Data Modification and Generation.

Standard Control Cards

All programs in the PRIME library with one exception (CHICHI) expect three standard control cards at the beginning of every problem deck, in the following order:

Title Card. All 80 columns of this card will appear at the beginning of the printed output from the program. This card must not be entirely blank.

Parameter Card. Columns 1-25 of this card are read and printed by the program. The fields and their meanings vary from one to another program.

Format Card. This card begins and ends with parentheses. A special coding system is used to describe the arrangement of fields on the data cards to be processed. All score fields are read in "real" mode (F specifications). The letter "X" is used to indicate skipped columns. If the program is one which can produce punched output for subjects, an "A" field specification must appear for the first (subject identification) field. An example would be (2X, A5, 3X, 7F2). The subject ID block is in columns 3-7 of the data cards, and the seven score fields, each two columns long, occupy columns 11-24.

If the program is not one that can produce punched output for subjects, only score-field specifications may appear in the format. An example is

(8X, 3F5.2, 5X, 14F1). Three fields, columns 9-23, are to be picked up first. Each of these is five columns long, with the last two columns representing two decimal places. Then five columns are skipped, and 14 more scores are picked up from columns 29-42. When the field is a whole number, only its length need be designated.

Group Control Cards

Three programs in the PRIME library involve procedures requiring physical subdivision of the data deck into groups of subjects. A particular control card is required in front of each sub-deck. In columns 1-5 of the card is punched the number of subjects in the sub-group to follow. The rest of the card may contain any titling information desired. The contents of these cards will appear in the printed output from the program.

Option Control Cards

Various of the PRIME programs allow or require the user to specify some kind of option for each of a series of variables, using a special control card. In such cards, each column is used to specify the option for one variable in the input list, with column 1 = variable 1, column 2 = variable 2, etc., continuing the list onto another card with column 1 = variable 81, if necessary.

Input Deck Arrangement

The input deck for a PRIME program consists of three sections: (1) system control cards, (2) the Fortran driver program and any override routines, and (3) one or more problem decks, containing data cards and PRIME control cards. An example is outlined below.

ABCD123, NAME.

XYZ = PASSWORD.

```
JOB, TM = 100.
EXECPF, 7037, PRIME.
789
      PROGRAM EXAMPLE (INPUT, OUTPUT)
      CALL UNITAB
      END
789
(Title card)
(Parameter card)
(Format card)
(Data cards)
(Blank card)
(EOF card)
```

User Identification Card

The first card in the deck may be obtained pre-punched at any terminal. The user enters his problem number and name.

Password Card

This also may be obtained pre-punched at any terminal. The user enters only his three-character password.

Job Card

A series of optional fields separated by commas is punched beginning in column one. The first field is the word "JOB." Other fields may appear in any order. All have default values.

TM is the time limit in seconds. For example: TM = 50. The default value is seven seconds. Since at present PRIME requires eight seconds of "overhead," this default should not be used.

PR is the print limit in pages. For example: PR = 50. The default value is 16 pages.

PU is the punch limit in cards. For example: PU = 103. The default value is zero (no punch). This value should be at least three more than the expected number of punched cards because separators are automatically included in the output (and should be discarded).

Priority of the job in the system is a function of the size of the program and the time limit. PRIME programs vary considerably in size and speed. Some indication of program size can be obtained by comparing the "RFL" values given in the program descriptions. Speed, of course, varies with the size of the problem to be solved, as well as the program.

PRIME Macro Card

This is punched exactly as shown in the example, beginning in column one. It serves to call up the PRIME library, compile the driver program and any override routines, and execute the assembled program.

789 Cards

One of these is placed before and after the driver program (or after the last override routine). Positions 7, 8, and 9 are punched in column one.

Driver Program

This is a Fortran program which does nothing but call up the PRIME program subroutine to be used. Each card begins in column seven, but spacing after that is unimportant (through column 72 only).

The name chosen for the program itself is arbitrary, but with a limit of seven characters. The name of the program must not be the same as that of the PRIME routine called, nor may it be the same as any of the

utility routines in PRIME. In addition to the program and utility routines, some of the programs employ "subsidiary" routines, which are unique in purpose. The names of the routines are listed in the program descriptions so that they may be avoided as names of the driver program as well. The easiest way to avoid proscribed names is to make the program name seven characters long.

The "buffers" whose names appear in parentheses, separated by commas, vary from one to another of the programs. All use INPUT and OUTPUT; some use PUNCH and/or TAPE2 also.

If any Fortran override routines are to be used, they would follow the driver program immediately -- before the second 789 card. Multiple override routines may follow the driver program in any order.

Problem Deck

Except with Program CHICHI, the input deck that is read by the program always begins with the three standard control cards: Title, Parameter, and Format. Some programs require or permit additional control cards before, after, or subdividing the data cards. Independent problem decks, each beginning with the standard initial control cards, may be stacked in one run. Limits on the job card apply to the aggregate of all stacked problems.

Blank Card

To allow stacking problem decks, control in the program returns to the beginning when a problem has been completed, and an attempt is made to read the Title card of a new problem. If the card read is entirely blank, the program terminates normally.

EOF Card

This is available pre-punched at any terminal. EOF stands for "End of File." These cards are usually green.

Standard Options

Missing Data Options

All programs in the PRIME library that process raw data, except those for cluster analysis, permit the automatic exclusion of missing data. There are three standard options available:

0 = both zero and blank fields are valid

1 = neither zero or blank fields are valid

2 = zero fields are valid, but blank fields are not

A particular column of the Parameter card is designated for this option. If one of the above is punched in this column, that option will be applied to all variables in the input list. If a "3" is punched in this column, however, an option control card (described in an earlier section of this manual) will be expected to follow the Format card. On this card the option desired for each variable is separately specified.

The manner in which missing data are excluded varies among the programs. Some exclude subjects for each variable considered separately, others consider the variables in pairs, and still others require valid data on all variables. Programs which do multivariate analyses based on intercorrelation matrices require complete data to ensure the integrity of the correlation pattern. However, it is possible to use Program MISCOR to obtain a punched correlation matrix based on pair-wise deletion of cases. This can then be entered instead of raw data to the

multivariate analysis program.

Output Options

Unless otherwise specified in the program description, there are four standard output options available to be entered in particular columns of the Parameter card:

0 = no output

1 = print only

2 = punch only

3 = both print and punch

Formats used in the programs for punched output for individual subjects are included with the program descriptions. See the description of subroutine ARRAY for standard punch formats for vectors and matrices.

Reporting Statistics

The PRIME programs report most statistics to four decimal places. Seldom is implying such precision warranted in research reports. Here are a few suggestions about reporting particular statistics:

Generally, descriptive statistics should be reported with only one digit more than the largest data value. If the input scores were two-digit whole numbers, for instance, a mean of 30.9806 and a sigma of 4.2237 should be rounded to 31.0 and 4.22.

Round F-ratios, chi-squares and correlation coefficients to two decimal places. Probability values should be reported only to the first non-zero digit (e.g. .0027 = .003, and .0567 = .06) and then only if they are .10 or less. Larger probabilities should be reported as "ns." If a program prints a probability as .0000, it could be reported as <.00005; strictly speaking a chance probability can never be zero.

Data Modification and Generation

All programs in the PRIME system except CHICHI and RELATE, call a subroutine named READX to acquire data from each source (usually subject). The reason for using a subroutine rather than a direct READ statement is to allow the user to override (replace) the standard version, which does nothing but read cards, with his own version. This permits data to be acquired from tapes or permanent files, and also permits the user to modify and generate data before they are sent on to the program.

Reading Data from Tapes or Permanent Files

Before we discuss the arguments to READX, consider this override version of the subroutine:

```
SUBROUTINE READX (ID, X, K, KF, N)
DIMENSION KF(10), X(N)
IF (K .EQ. 1) REWIND 2
READ (2, KF) X
RETURN
END
```

The REWIND can be done on the first call, or in the driver program, or with a system control card. With programs that use Group control cards, it should not be done on the first call, for reasons to be explained later.

If the data were stored on permanent file 5432, and the sub-file to be used was named DATA, this system control card would be inserted just before the EXECPF card that calls PRIME:

```
READPF, 5432, DATA = TAPE2.
```

Care must be exercised with programs that use TAPE2 as a dedicated scratch

unit; some other unit number would have to be used for the input file.

The buffer name TAPE2 would be added to the list in the driver program, and the cards for the override READX would be inserted immediately after the driver program, before the second 789 card.

If the data were stored on tape, rather than on a permanent file, the following pair of cards, instead of the READPF card, would be used:

```
RFL (00000)
REQUEST, TAPE2, 5432/0000, RO.
```

The number is parantheses on the RFL card would be obtained from the program description in this manual.

Arguments to INPUT

Argument ID is irrelevant, except with programs that can produce punched output for individual subjects. Such programs expect an A-mode subject identification field to be returned along with the scores.

Argument X is the vector in which scores are returned from READX to the calling program.

Argument K comes from the calling program. It will be the serial order number of the subject in the sample, since READX is called once for each subject. However, in programs that use Group control cards, K will begin again at "1" as each group is started.

Argument KF comes to READX containing the contents of the Format control card (10A8). In some applications the format to be used can be included in READX and the Format card can be used to transmit other information to READX.

Argument N comes to READX as the number of scores to be returned in vector X. In applications where data are to be generated or com-

bined in READX, this value may be larger or smaller than the number of of scores actually read.

Modifying Data

Suppose that a data set was punched using the code "99" to indicate missing data. Since the PRIME programs cannot recognize this convention, one could convert the 99 codes to the equivalent of reading a blank field numerically (-0.0), using this version of READX:

```
      SUBROUTINE READX (ID, X, K, KF, N)
      DIMENSION KF(10), X(N)
      READ KF, X
      DO 2 I = 1,N
    2 IF (X(I) .EQ. 99) X(I) = -0.0
      RETURN
      END
```

This is, of course, only one of an infinite number of ways in which READX can be used to change the data before it gets to the program.

Generating New Variables

Suppose that we have two scores punched on each data card, and we want to analyze them. But we also want to analyze the difference between them, as well as their ratio.

```
      SUBROUTINE READX (ID, X, K, KF, N)
      DIMENSION KF(10), X(N)
      READ KF, X(1), X(2)
      X(3) = X(1) - X(2)
      X(4) = X(1) / X(2)
      RETURN
      END
```

The value of N, the number of scores to be returned and analyzed, which comes from the Parameter control card, will be four in this example. The Format card, however, would specify only two score fields.

This technique is especially useful in regression analysis where product variables are needed to register interactions and powered variables are needed to register curvilinearity.

In some of the more esoteric applications, the programmer may wish to communicate between the driver or other routine to READX through COMMON. Blank common should be avoided for this purpose, since it is used in many of the PRIME program routines.

Test Scoring and Item Analysis

Two test scoring and item analysis programs are included in PRIME. The SCALES program provides Likert scaling of mutually exclusive one-digit item sets or right-wrong scoring of comparable sets of item-choice numbers. The RASCH program provides Rasch model analysis of item data coded 1 = right, 0 or blank = wrong. Both programs provide item analysis (calibration) information but of rather different kinds. Both programs also permit punched output of composite scores for individual subjects.

Program SCALES

This routine is intended to handle two very common data-processing chores: scoring of multiple-choice tests and scaling -- the calculation of Likert (unit-weighted) scale scores. Item analysis information is also provided. An option permits punched output of one card per subject, containing either right-answer count(s) or Likert scale scores.

Input Deck Arrangement

Title card

Parameter card

Format card

(Scale key card)

(Reversal or right-choice key card)

Data cards

Parameter Card Fields

1-5 = number of items (max. = 300)

6-10 = number of subjects

15 = number of scales (max. = 9)

20 = scaling/scoring option

0 = scaling with no reversals

1 = scaling with item reversals

2 = right-answer scoring

25 = 1 for punched output of scores

Input

Item data is limited to one-digit positive integers. Zero and blank fields will be considered missing data for scaling.

The first field specification in the format must be subject identification (max. = A8).

Scale key card. This card is required when the number of scales indicated in Parameter column 15 is greater than zero. If Parameter column 15 is zero, one scale containing all items is assumed and this card is omitted. This is a standard option card and contains for each item in turn the number of the scale to which the item is to be assigned. Un-

used items can be coded zero on this card. Each scale must contain at least two items. Although each item can be assigned to only one scale, READX could be used to extend the item list with duplicates.

Item-scale reversal key card. This standard option card is required if any of the item scales are to be reversed (e.g. 1 = 5, 2 = 4) in the scale-scoring process. This is sometimes necessary so that high item scores will have a common meaning. This card is required only when Parameter column 20 = 1. For each item that is to be reversed, the number of scale points (the highest possible score) is entered in the appropriate column. The rest of the card is left blank.

Right-answer key card. This standard option card is required if Parameter column 20 = 2. For each item in turn the correct choice number is entered. Use of zero or blank will result in blank items being scored correct.

Output

Scale statistics are printed on the first page, followed by a table of item analysis data.

Scale statistics. For each scale, the number of items, the scale score mean, the scale score sigma (parameter value), and the alpha coefficient of internal consistency reliability (Cronbach, 1951) are provided. These computations will be based only on subjects who have no missing data. For cognitive measures alpha should be at least .90; for personality or attitude measures .70 is usually considered adequate.

Item statistics. The first six columns of this table contain (1) the item number, (2) the scale to which it is assigned, (3) the reversal key, (4) the item mean, (5) the item sigma, and (6) the correlation of

the item with the total score of the scale to which it was assigned. Since the item is included in the scale total, these correlations are somewhat inflated, but to a common degree. Only subjects with complete data are used for these computations.

The last 10 columns of the table contain frequency distributions for each item, prior to any reversals that are to be carried out. Blank items are coded zero for this table.

Punched Output. If Parameter column 25 = 1, one card will be punched for each subject with this format: (A8, F2, 9F5). The A field will contain the subject identification field of the input format, the F2 field will be the constant " 1," and the scale scores or right-answer counts occupy the rest of the card.

Although the statistics are based only on subjects with complete data, scale scores will be prorated for missing data (provided at least half the items of a scale are answered), and a card will be punched for every subject.

Buffers

INPUT, OUTPUT, PUNCH

RFL = 27000 with a TAPE added.

Program RASCH

This program is designed to carry out a test calibration and scoring procedure described by Wright (1977) for applying the logistic response model developed by Georg Rasch (1960). The estimates of item difficulty are not dependent on the ability level of the calibration sample, nor are the estimates of person ability dependent on the calibrated difficulty of the items used. The technique is therefore

especially useful for "tailored testing." Another outcome of the calibration process is a set of "residual fit statistics" which reflect the degree to which the data conform to the expectations of the model. These are useful in detection of item bias, ambiguity, and other faults. Fit statistics can be computed for each subject as well as for each item and for the whole test.

Input Deck Arrangement

Title card

Parameter card

Format card

Data cards

Parameter Card Fields

1-5 = number of items (max. = 300)

6-10 = number of subjects

15 = output option for difficulty and ability estimates

20 = 1 for punched output for subjects

Input

The Format must include an A field specification for subject identification before item-score fields are specified.

The input data are expected to be item scores coded 1 = right, 0 = wrong. If conversion of choice numbers to right/wrong codes is necessary, it can be done in READX with a key card read in on the first call.

Because the Rasch model considers zero and perfect scores to be indeterminate with regard to the measurement of ability, such records are edited out of the data at the start of the program. The same is true of any item that is never or always answered correctly.

Output

A list of frequencies of right answers for each item is labeled ITEM. A list of frequencies for each possible score (count of right answers) is labeled SCORE. Next is printed a list of the item difficulty estimates, labeled ITEM D. These are scaled such that the average of the item difficulties is zero if they are symmetrically distributed. Standard errors, labeled SE OF D, are provided for the difficulties also. Ability estimates corresponding to each possible right-count score are labeled ABILITY. These are symmetrically distributed around zero. Standard errors for these estimates are also provided.

The final list in the printed output, labeled RMSQ, contains a residual mean square value for each item. This statistic has an expected value of 1.0 when the item responses of the subjects correspond to their ability estimates and the item difficulties as predicted by the Rasch model. Items with RMSQ values as large as 3.0 would be suspect in terms of their appropriateness within the set. Finally, an RMSQ value is computed for the complete test.

Punched output of one card per subject is an option in this program. Such cards have the format (A8, F2, 3F10.4). Subject identification from the input format appears in the A field, the F2 is the constant " 1," and the three scores are: (1) count of right answers, (2) the corresponding ability estimate, and (3) a RMSQ value indicating the degree to which the subject's responses fit the Rasch model.

Buffers

INPUT, OUTPUT, PUNCH, TAPE2, TAPE3

RFL = 45000 with a TAPE added.

Subsidiary Routines

EDIT, PASS, DEEBEE, RESIDS

Descriptive Statistics and Chi-Square

The three programs described in this section are designed to provide frequency tabulation and analysis. The UNITAB program computes means and sigmas and offers compact output of frequency distributions for a series of variables. The DUOTAB program compiles two-way cross-tabulations and computes chi-squares for any or all pairings of a series of variables. The CHICHI program performs one or two-way chi-square analyses of frequency data already compiled by other means.

The author strongly recommends the use of the UNITAB program with every newly-punched data set to detect coding and/or punching errors, to detect badly distributed variables, and to provide a general sense of the centrality and range of the measures.

Program UNITAB

This program is designed to provide descriptive statistics and frequency distributions for a series of variables. Although the program expects the data to consist of one- or two-digit positive whole numbers, data scaled otherwise can be modified in READX, or in a special routine named ARRAY which is also called during the data-reading sequence.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

(Distribution option card)

Data Cards

Parameter Card Fields

1-5 = number of variables (max. = 150)

6-10 = number of subjects

15 = missing data option

20 = distribution option (0, 1, or 3)

Input

A missing data option card is expected after the Format card only if Parameter column 15 = 3. A distribution option card is expected only if Parameter column 20 = 3. A distribution option of zero will result in the suppression of the printing of frequency distributions. Only score fields (no subject ID) may be specified in the Format card.

The input data are rounded automatically to two-digit integers before tabulation, but the original values are used to compute the descriptive statistics. If data are modified in READX, descriptive statistics will be computed from the modified values. To modify data after computing statistics but before tabulation, an override version of a dummy routine named ALTER may be added to the driver program. This routine is described elsewhere in this manual.

Output

For each variable the mean, sigma (population parameter), standard deviation (sample statistic), number of valid scores, and percentage of valid scores are printed. If distribution option one is used, a tabulation of frequencies and percentages of all occurring scores is also printed. The output is blocked with 10 variables across the page. A

non-zero frequency for any of the 10 variables will force printing of the line of frequencies for that score value.

Buffers

INPUT, OUTPUT

RFL = 66000 with a TAPE added

Program DUOTAB

This program provides cross-tabulation of any or all pairings of a series of variables. A chi-square value is also computed for each frequency table. Data are automatically rounded to two-digit positive whole numbers before tabulation, but READX can be used to modify the scores if necessary to fit this restriction.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

(Cross-tab selection card)

Data cards

Parameter Card Fields

1-5 = number of variables (max. = 99)

6-10 = number of subjects

11-15 = number of cross-tab selections (max. = 20)

20 = missing data option

Input

The Format card may include only score-field specifications. All scores will be rounded to two-digit positive whole numbers before tab-

ulation. Missing data on either variable of a particular pair will exclude the subject only from that pairing.

Cross-tab selection. If Parameter column 15 = zero, all non-redundant cross-tabs will be computed and no cross-tab selection card is expected. The number in Parameter column 20 is the number of cross-tab specifications expected to appear on a special control card. Each cross-tab is indicated by a pair of two-digit variable numbers; the first selection occupies column 1-4, the second column 5-8, etc. If the second variable number of a pair is zero, the first variable will be cross-tabbed with all others in turn. This is considered only one selection, however, for the count in Parameter column 15.

Output

Each cross-tab frequency table will contain a row or a column for each unique value that appears in the sample for the variables concerned. Row and column sums and a table total are printed in addition to the cell frequencies.

A chi-square value is automatically computed for any table which has more than one row and more than one column, along with its chance probability. The continuity correction is applied with two-by-two tables. No check is made regarding the assumption that all expected frequencies are at least five. If this is not met, rows and/or columns could be combined in READX, or the combined frequencies could be submitted to the CHICHI program.

Buffers

INPUT, OUTPUT, TAPE2

RFL = 102000 with a TAPE added

30

Subsidiary Routine

CROSS

Program CHICHI

This is the only program in the PRIME system that does not employ the standard initial control cards, nor does it analyze raw data from individual subjects. A fixed format is used to enter control information and the data, which are cell frequencies. Problems may be stacked as with other PRIME routines, and a blank card follows the last problem, as usual.

Data Card Fields

1-6 = alphameric problem label

7-8 = rows of frequency table (max. = 50)

9-10 = columns of frequency table (max. = 99)

11-80 = up to 14 five-digit frequencies

Input

If more than 14 frequencies are to be entered, continue the list on subsequent cards, beginning each in column 11.

For two-way tables, frequencies are entered across rows of the table, continuously. Only cell frequencies, not marginals, are entered. For a two by three table, for example, this would be the order of entry:

1	2	3
4	5	6

For one-way tables (lists), two options are available: (1) If the table is described in columns 9-10 as having one column, the observed frequencies will be tested against a null hypothesis of equal frequencies in all cells. (2) If the table is described in columns 7-8 as having one row, the observed frequencies will be tested against a list of specific expected frequencies which occupy a separate card with format (10X, 10F7.2) that follows the observed frequency card(s).

Although data are expected from cards, a permanent file, properly formatted, could be equated to INPUT.

Output

For each problem, the program prints observed and expected frequencies, row and column sums, total N, chi-square, degrees of freedom, and the chance probability value.

No check is made of the assumption that each expected frequency is five or more.

Buffers

INPUT, OUTPUT

RFL = 52000

Transformations and Composites

The programs in this section produce two kinds of output scores. Program REFORM converts scores on a series of variables to one of three types of standard scores: Z scores, percentiles, or T scores. In addition it prints tables of equivalents for all three conversions. Program SCORES computes weighted composite scores using raw data and weights which are normally punched by one of the multivariate analytical programs in PRIME.

Program REFORM

Conversion of raw data to standard score form has a number of advantages, depending on the nature of the transformation chosen. All three conversions convert the data to a common (F5.2) field specification which may greatly simplify subsequent analyses. Percentiles are more easily interpreted by non-statisticians. Normalized standard scores will better fit the assumptions of many analytic procedures than will raw scores with skewed distributions. The printed tables of equivalents are useful for developing a manual for a new instrument.

The transformations built into Program REFORM require data from the normative subject sample. There are many possible non-linear transforms that can be carried out on a within-subject basis in subroutine READX without resorting to a special program to convert the data.

It is important to realize that the normalizing transformation will do little to improve a case of severe truncation (as opposed to skewness). In most such situations it would be better to convert the raw scores to binary or trinary form to achieve distribution symmetry.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards

Parameter Card Fields

1-5 = number of variables (max. = 25)

6-10 = number of subjects

15 = missing data option

20 = punched output option

0 = no punched output

1 = Z scores ($\mu = 50$, $\sigma = 10$)

2 = percentiles

3 = T scores (normalized Z)

25 = 1 to print tables of equivalents

Input

The first field specification in the Format must be subject identification (max. = A8).

Output

For each variable the number and percentage of valid cases, the mean and the sigma will be printed. The equivalents table that follows (unless suppressed) contains six columns: (1) raw score value, (2) frequency, (3) percentage, (4) percentile equivalent, (5) standard score (Z) equivalent, and (6) normalized standard score (T) equivalent.

Percentile conversion shifts the shape of the distribution as close to rectangularity as possible. Standard scores are a linear transform which does not affect distribution shape. Normalized (T) standard scores will be distributed as close to the normal (bell-shaped) curve as possible.

The raw-score column of the equivalents table will contain four-place decimal fractions even when integers are input. This is because the program divides the data range into up to 1000 intervals, compiles frequencies for each, and then estimates the mid-points of intervals in which one or more raw scores occurred and prints those. In almost all cases,

rounding these values to the number of input digits will identify the input value without ambiguity.

Punched output will be in the format (A8, F2, 14F5.2), where A8 = subject identification from the input card, F2 = the constant " 1," and the transformed scores follow. Missing data are omitted separately for each variable and will be punched as zero (invalid) on output cards.

Buffers

INPUT, OUTPUT, PUNCH, TAPE2

RFL = 116000 with a TAPE added

Program SCORES

This program is designed to compute and punch one or more weighted composite scores, using a set of weights and raw data as input. The general form of the equation used is this:

$$C = a + b_1X_1 + b_2X_2 + \ldots + b_KX_K$$

In this equation the X symbols represent the K variables, and the b symbols represent their respective weighting constants. The a is an additive constant. The weights may either be for raw scores, or for z-scaled X variables (beta weights). If the weights are for standardized ($\mu = 0$, $\sigma = 1$) data, the a coefficient will be zero. When raw score weights are used, a may or may not be zero. Likert scales (see Program SCALES) are thus a special case where a = 0 and all the b = 1.

Normally, the weights will be obtained as punched output from one of four analytic programs in PRIME: REGRAN, CANONR, MANOVA, or FACTOR. However, with properly formatted weight cards, the program can be used for any problem requiring computation of weighted composite scores.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

(Mean and sigma cards)

Weight cards

(Criterion variable list)

Data cards

Parameter Card Fields

1-5 = number of variables (max. = 100)

6-10 = number of subjects

11-15 = number of composites (max. = 25)

16 = missing data option

17 = 1 to use raw score weights

18 = 1 if weights are punched by columns (REGRAN only)

19 = 1 to compute residual scores (REGRAN only)

20 = 1 to read means and sigmas

Input

The first field specification in the Format card must be for subject identification (max. = A8).

Standardization. Programs FACTOR and CANONR (always) and Program REGRAN (optionally) produce weights for z-scaled data. If Parameter column 17 = 0, the program will convert the raw-score data to standard scores ($\mu = 0$, $\sigma = 1$) before computing composites. However, if Parameter column 20 is also = 1, the means and sigmas used will be those read in before the weight vectors, rather than those computed from the current

data set. The format for these cards, which can be obtained from the analytic program that punches the weights, is (10X, 7F10.4). Weight cards are also expected to be punched in this format.

Missing data. Substitution of the sample mean is used to replace missing data before computation of composite scores.

MANOVA problems. Weights for discriminant scores are in raw score form with no a value, punched by rows (variables). Parameter columns 17-20 = 1000.

FACTOR problems. Weights are for standard scores, punched by rows (variables). Parameter columns 17-20 = 0000 or 0001.

CANONR problems. Although input to CANONR is a single list of A variables followed by B variables, separate weight decks are punched for sets A and B, and two problem decks, weights and data, must be entered to obtain canonical scores for A and B. The weights are for standard scores, punched by rows. Parameter columns 17-20 = 0000. If Parameter column 20 = 1, the mean and sigma cards would have to be repunched for the second problem deck, unless set A is a multiple of seven variables.

REGRAN problems. REGRAN can produce either raw or standardized weights, punched by columns (models). Raw score weight lists will have a regression constant (a) at the end of each, but this is not included in the count of variables.

If Parameter column 19 = 1, instead of punching predicted criterion scores for the series of models as a list, the program will first subtract the predicted score from the subject's actual criterion score for each model. If this option is used, a list of criterion variable numbers punched in format (25F2) must be entered after the weight vector cards.

Output

Printed output will include means and sigmas if they were to be computed for standardization.

Punched output will be in the format (A8, F2, 14F5.2) if standard-score weights (Parameter col 17) are used, or (A8, F2, 7F10.4) if raw-score weights are used. The F2 field is the card number within a subject's set. One or more cards containing composite scores will be punched for each subject in the input sample.

Buffers

INPUT, OUTPUT, PUNCH, TAPE2

RFL = 42000 with a TAPE added

Cluster Analysis

The purpose of the procedure implemented by Programs GROUPS and ASSIGN is to identify subsets of objects (usually people) which are more similar to each other than they are to objects in other subsets. The term "similar" is defined in terms of profiles or patterns of scores on one or more variables. The numerical index of similarity employed is the Euclidean distance function -- the sum of squared differences of scores for two objects across the series of variables.

The method used in Program GROUPS is called Hierarchical Grouping Analysis (Ward, 1963). It is a stepwise procedure which begins with N one-person "groups" and proceeds to combine the most similar profiles of group means. Normally only the last ten or so steps in this process are of interest (ten down to two groups). At each of these last stages, the program prints the profiles of group means which have been established. Also printed is an "error increase" value which is an index of the amount

of within-group variance increase caused by the combining of the two groups that produced the current clustering. This is useful in locating an "optimum" number of groups.

Once a particular number of groups has been chosen, the program may be re-run to obtain punched output of the group-mean profiles. These and raw data may then be submitted to the ASSIGN program which will compute the Euclidean distance index from each group and punch an output card for each subject. This can be carried out with the original subjects or with others not included in the grouping analysis. Once subjects have been assigned to the empirically defined groups they may be compared on other variables of interest.

Although the analytic process is a stepwise reduction from N groups to two, the output can be examined in reverse order as an unfolding "tree" structure with successively greater differentiation of subject clusters.

Although the procedure is empirical and objective, it is purely descriptive in nature. No significance test can be computed to estimate the likelihood of replication with a different sample. However, separate groupings of random subsamples can sometimes lend credibility to an obtained set of profiles.

Program GROUPS

This program carries out the stepwise clustering process described earlier, after first converting the raw data on each variable to standard score form ($\mu = 0$, $\sigma = 1$). This ensures that all variables in the profile will contribute equally to computation of the distance function.*

Normally, the program is first run with a "number of groups" value of ten or so. The printed output is examined to determine an "optimum"

*A minus sign in Parameter column 11 will suppress this conversion.

number of groups, and then the program is rerun to obtain punched output of that number of profiles of group means. The raw data and these profiles are then input to the ASSIGN program to obtain cards indicating distances of the subjects from each group profile.

Input Deck Arrangement

Title card

Parameter card

Format card

Data cards

Parameter Card Fields

1-5 = number of variables*

6-10 = number of subjects*

11-15 = number of groups (stage to begin output: KP)

20 = 1 to punch KP group profiles

25 = output option for variable means and sigmas

*The value NS (NV + 7) may not exceed 20,000.

Input

The format must include only specifications of score fields. The numbers of variables and subjects are limited only in terms of the total number of scores, but processing of large numbers of subjects will be relatively slow.

Output

Printed output will begin at the stage (number of groups) designated on the Parameter card. At that stage and all successive stages down to two groups, profiles of group means in standard-score scaling will be printed. The initial print line for each stage indicates the pair of

groups which were last combined and the increase in "error" (within group variation) caused by that combination.

The group numbers (labeled "G") are not serial, since at each stage of the reduction process the smaller group number of the two combined is retained. The number of cases in each group is labeled "N" and is followed by a list of group means for each variable in the profile.

If Parameter column 20 = 1, cards containing the profiles of means will be punched in format (A5, 2F2, 1X, 14F5.2) at the stage (KP) indicated on the Parameter card. In this format the A5 is the word "GROUP," the first F2 is the group number, and the second F2 is the card number. Although this program allows designation of any number of groups, the ASSIGN program is limited to a maximum of 14. The group numbers punched in these cards are serially ordered, unlike those in the printed output.

Buffers

INPUT, OUTPUT, PUNCH

RFL = 76000 with a TAPE added

Subsidiary Routine

GROUP

Program ASSIGN

This program was designed for use in conjunction with Program GROUPS and the punched output of group profiles it produces, but it can be used in any situation where one wishes to classify subject profiles with regard to a series of standard profiles, which could be defined on a theoretical rather than empirical basis.

The standard or criterion profiles are assumed to be z-scaled values ($\mu = 0$, $\sigma = 1$). Subjects' score profiles are converted to z-score form,

either on the basis of the sample itself, or using means and sigmas provided to the program along with the criterion profiles.

A Euclidean distance value is computed for each subject against each of the criterion profiles. For each criterion this is the square root of the sum of squared differences between corresponding criterion and subject scores across the variables of the profile, divided by the number of variables. The smaller the value, the closer the subject is to that criterion profile.

$$D_c = \sqrt{\frac{\sum_{1}^{K}(z_1 - z_c)^2}{K}}$$

Punched output from the program is one card per subject containing the distance measure from each criterion profile and the index number of the smallest (closest) one.

Input Deck Arrangement

Title card

Parameter card

Format card

(Variable mean and sigma cards)

Group mean profile cards

Data cards

Parameter Card Fields

1-5 = number of variables (max. = 99)

6-10 = number of subjects

11-15 = number of groups (max. = 14)

20 = 1 if variable means and sigmas are to be read rather than computed from the sample

Input

The groups-by-variables profile matrix is input by rows, each row (group) beginning on a new card. The format used to read these cards is (10X, 14F5.2). Punched output from Program GROUPS is appropriately formatted. If means and sigmas for the variables are to be read in, the format used is (10X, 7F10.4), also as punched by Program GROUPS.

The Format control card must include a subject identification field (max. = A8) as the first field specification.

Output

Group mean profiles are printed by the program for reference purposes. Variable means and sigmas are also printed.

Punched output is one card per subject in the format (A8, F2, 14F5.2). The F2 value is the index number of the closest criterion profile.

In most applications the number of subjects assigned to each group may differ slightly from the numbers indicated in the output of Program GROUPS. The reason for this is that during the grouping process the "centers" of the groups shift to some extent, but subjects remain assigned to the groups in which they were initially placed. Normally, only borderline cases will be affected by this shifting.

Once a sample of subjects has been assigned to type groups, it would be possible to "purify" the groups by using only cases with the smallest D values in comparisons of the types on other variables.

Buffers

INPUT, OUTPUT, PUNCH, TAPE2

RFL = 42000 with a TAPE added

Analysis of Variance

Although usually viewed as a method for answering questions about differences between group means, this family of procedures attains such answers by partitioning the variance of one (univariate) or more (multivariate) dependent variables into systematic variance associated with one or more independent variables and that remaining, which is considered random or error variance. The ratio of systematic to error variance is tested for significance with an F-ratio.

Independent variables are nominal scales, consisting of a set of categories or levels, which have no ordered quality, even if defined as "high, medium, and low" in some respect. There are two types of independent variables: Between-subjects variables are those that classify people (or other source of the dependent measures), e.g. sex or experimental group. Within-subjects variables are those that classify repeated measures of the same people on the dependent variable, such as pre and post tests.

Program AOV123 will handle one, two, or three factor (independent variable) designs with any combination of between and within factors. For a particular design and sample of subjects, a series of dependent variables may be entered and each will be independently analyzed in turn.

Multivariate analysis of variance is an extension of a one-between-factor design in which a series of dependent variables is used together and differentially weighted to discriminate between the levels (groups) of the independent variable. With more than two groups, the discrimination may be multidimensional, potential independent dimensions being one less than the number of groups. The MANOVA program also produces univariate

anova results for each dependent variable.

The third program described in this section, INTRAR, uses analysis of variance to compare between-subjects variance to within-subjects (raters) variance to produce reliability estimates called intraclass correlations.

Program AOV123

This program computes an unweighted means, fixed effects analysis of variance with equal or unequal cell N for one, two, or three factor designs with any combination of between and within subjects factors for each of a series of dependent variables. Computational formulas may be found in Kirk (1968).

With unequal cell N, the user should be aware of the assumptions of the procedure. Computations are done to estimate what would have been obtained if cell N had been equal. Thus, data are assumed to be "missing" for random, or at least irrelevant, reasons. With research questions derived from theory, this assumption is usually reasonable. When inferences are to be made about particular extant populations, however, a "least squares" (regression) analysis would be preferred.

For each "effect" (main effect or interaction) an F-ratio and associated chance probability is computed to reflect the reliability of the observed mean differences. A value called "omega squared" is also computed to reflect the absolute size of the effect. A particular effect may be very significant statistically (small p value) and yet account for a trivial amount of variance in the dependent variable -- a common result with very large samples.

It is not always recognized that one-factor between and within designs with two groups yield results identical to what are called "t-tests." In this sense, then, Program AOV123 computes t-tests.

Input Deck Arrangement

Title card

Parameter card

Format card

(missing data option card)

Data cards (subdivided by Group cards)

Parameter Card Fields

1 = model number

2-5 = number of dependent variables (max. = 100)

6-10 = levels of factor A (max. = 15)

11-15 = levels of factor B (max. = 15)

16-20 = levels of factor C (max. = 15)

21 = missing data option

22-25 = number of subjects per between-subjects cell, if all cell N are equal. For unequal cell N, set = 9999 and use Group cards.

Input

The format should include only score-field specifications. The total number of scores per subject may not exceed 10,125. If the design incorporates a within-subjects factor, subjects with an invalid score on any level of that factor will be excluded from the analysis.

The model number in Parameter column 1 specifies the design to be assumed:

1 = BBB, all between-subjects factors
Data order = A(B(C(S(V))))

2 = BBW, third factor is within-subjects
Data order = A(B(S(C(V))))

3 = BWW, last two factors are within-subjects
Data order = A(S(B(C(V))))

4 = WWW, all factors are within-subjects
Data order = S(A(B(C(V))))

For two-factor designs, set col. 20 = 1. BB = model 1, BW = model 3, WW = model 4.

For one-factor designs, set both col. 20 and col. 15 = 1. B = model 1, W = model 4.

In the data-order notation above, A, B, and C refer to the factors of the design, S = subjects within a group, and V = dependent variables. The nested parentheses indicate the expected order of entry of scores, inner parentheses varying most rapidly. For example, consider this 2 by 2 by 2, model 2 (BBW) design, with two subjects per cell and three dependent variables, assuming two cards per subject. This is the expected order of the cards:

$(A_1\ B_1\ S_1\ C_1)\ V_1\ V_2\ V_3$
$(A_1\ B_1\ S_1\ C_2)\ V_1\ V_2\ V_3$
$(A_1\ B_1\ S_2\ C_1)\ V_1\ V_2\ V_3$
$(A_1\ B_1\ S_2\ C_2)\ V_1\ V_2\ V_3$
$(A_1\ B_2\ S_1\ C_1)\ V_1\ V_2\ V_3$
$(A_1\ B_2\ S_1\ C_2)\ V_1\ V_2\ V_3$
$(A_1\ B_2\ S_2\ C_1)\ V_1\ V_2\ V_3$
$(A_1\ B_2\ S_2\ C_2)\ V_1\ V_2\ V_3$
$(A_2\ B_1\ S_1\ C_1)\ V_1\ V_2\ V_3$
$(A_2\ B_1\ S_1\ C_2)\ V_1\ V_2\ V_3$
etc.

The format for this example could specify three score fields. All scores for a subject are read at the same time, so the format must be adequate to describe them all. If the format for all cards is the same, only one card description is needed, and the format will be automatically repeated. In other words (5X, 3F5) will operate the same as (5X, 3F5/ 5X, 3F5).

Output

If Group cards are used, their contents are printed. For each dependent variable a standard "source table" is printed, including sums of squares (S.S.), degrees of freedom (D.F.), mean squares (M.S.), F ratio (F), chance probability values (P), and omega-square values (O2). The error term used in computing the F ratio for each effect is also indicated.

Following the source table, means for all main effects and interactions are printed. For the highest order interaction, cell means and standard deviations (sample statistics), along with cell N are printed. When cell N are unequal, main effect means may differ from values computed by other programs, since all higher-order means are computed here as means of the cell means.

Buffers

INPUT, OUTPUT, TAPE2

RFL = 102000 with a TAPE added

Subsidiary Routines

AVBBB, AVBBW, AVBWW, AVWWW, AMONG, AVTAB, AVB11

Program MANOVA

Multivariate analysis of variance is an extension of the one-between univariate anova design which includes simultaneously a set of dependent

variables that are weighted to define a set of orthogonal composite variables (called "discriminant functions") in such a way that the multivariate means (called "centroids") of the groups of subjects are maximally separated from each other.

A statistic called "Wilk's Lambda," which is analogous to $1-R^2$ in regression analysis (and identical with two groups) is tested for significance. This test constitutes multivariate analysis of variance, as most authors define the term. The computation of discriminant loadings and centroids, which the program also accomplishes, is known as "discriminant analysis." Computational procedures are based on formulas provided by Cooley and Lohnes (1971).

Discriminant functions have much in common with factors in factor analysis. They define a "space" in which individual subjects as well as group centroids can be located. In fact, one of the options in the program is the punching of a set of weights that can be used with Program SCORES to obtain discriminant scores for individuals.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards, sub-divided by Group cards

Parameter Card Fields

1-5 = number of predictor (dependent) variables (max. = 75)

6-10 = number of subject groups (max. = 25)

15 = missing data option

20 = output option for discriminant weights

Input

The format must include only score-field specifications. The number of subjects in each group should exceed the number of variables. The missing data option will exclude any subject with one or more missing scores.

Output

The content of the Group cards is printed, with an indication of how many in each group had complete data and were used in the analysis. Wilk's Lambda and the test of its significance is printed. Since this is an index of non-predictive variance, small values will be associated with small (significant) probabilities. Following the F test of overall discrimination, each discriminant function (root) is tested for significance with a chi-square. The percentage of variance (PCT. V) associated with each root is also given.

A table of centroids is printed next; the mean on each function (column) is given for each group (row). A table of discriminant loadings appears next. These are correlations between each of the dependent variables (rows) and each of the functions (columns). Finally, univariate analysis results for each of the dependent variables separately are reported.

Weights for use with Program SCORES are usually not of interest, except to be punched for that purpose.

Buffers

INPUT, OUTPUT, PUNCH

RFL = 76000 with a TAPE added

Program INTRAR

Program AOV123 computes fixed-effects analysis of variance. There is one application of a random-effects analysis (where levels of the independent variable are a sample of all possible levels) which is especially useful. This is when data are obtained on a set of subjects from multiple sources, such as a group of raters. The subjects are treated as levels of an independent variable, and the raters are the source of scores for each subject "group."

The F-test of significance, as usual, compares between-subject (numerator) to within-subject (denominator) variance. Instead of using omega-square to estimate the degree to which subjects are discriminated reliably by the raters, however, a similar coefficient, called an "intraclass correlation," is computed. If the raters agree with each other and the subjects are genuinely different on what was rated, the F ratio and the corresponding intraclass correlation will be large. A coefficient exceeding .70 is usually a sufficient indication of rater reliability.

Of practical importance is that this method of computing rater reliability does not require that the same judges rate all subjects, or even that the number of judges be equal for each subject. However, if the same judges are used, and they differ in the average rating they give, such variance is not removed from the error term as it would be if an ordinary correlation were used to estimate reliability.

Computing formulas for this procedure may be found in an article by Ebel (1951).

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards

Parameter Card Fields

1-5 = number of variables (max. = 100)

6-10 = number of subjects

11-15 = number of raters

20 = missing data option

Input

The format must specify only score fields. The number of raters x number of variables may not exceed 5000 for any subject. The data input order is S(R(V)). Normally, the ratings of each rater-subject combination on the series of dependent variables are punched on a separate card. For instance, this would be the card order for a problem with three subjects, two raters, and five variables:

(S 1 R 1) V 1 V 2 V 3 V 4 V 5

(S_1 R_2) V_1 V_2 V_3 V_4 V_5

(S_2 R_1) V_1 V_2 V_3 V_4 V_5

(S_2 R_2) V_1 V_2 V_3 V_4 V_5

(S_3 R_1) V_1 V_2 V_3 V_4 V_5

(S_3 R_2) V_1 V_2 V_3 V_4 V_5

All scores for one subject are read at the same time. If the card format is the same for all raters, only one card need be described and the

format will be automatically repeated, e.g. (10X, 5F1) for the example.

If the number of judges per subject is unequal, indicate the largest number on the Parameter card, fill out the deck with blank cards, and use the missing-data option to skip them.

Output

In the printed output table, each dependent variable is represented by a separate row. The column headed "K" is the number of raters per subject, estimated when this is unequal per subject. The "P" column is the chance probability derived from the F ratio, which applies to both the RA and RI coefficients.

Two different intraclass correlations are computed for each variable. The RA value estimates the reliability of average ratings by K judges. The RI coefficient estimates the reliability of ratings made by individual judges.

When the F ratio is less than 1.0, the intraclass correlation is negative. In the output from this program, such negative values are arbitrarily set to zero.

Buffers

INPUT, OUTPUT

RFL = 37000 with a TAPE added

Intercorrelation Analysis

The Pearson product-moment correlation coefficient is an index of the relationship between two variables. Its square reflects the proportion of the variance of the variables which is common or shared.

The basic computing formula, when applied to particular types of data, is often given a special name. When both variables are ordinal,

it is called a "rank-order" correlation or "Rho." When one of the variables is a dichotomy and the other is an interval scale, it is called a "point-biserial" correlation. When both variables are dichotomous (binary), the result is called a "phi coefficient."

Matrices of correlations of all combinations of a set of variables are the basis for a number of multivariate procedures, such as canonical correlation, multiple regression, and factor analysis. In this section we will describe two programs which produce matrices of intercorrelations and cross-set correlations. We will also describe a program for canonical correlation, which is a multivariate method closely related to multiple correlation. We are including it here because in practice it is most often used to "summarize" a matrix of cross-set correlations.

Programs MISCOR and SELCOR provide pair-wise deletion of missing data, establishing a sample of valid cases independently for each correlation. Programs CANONR, REGRAN, COMMAP and FACTOR all use row-wise deletion, using only subjects who have complete data on all variables. These programs all begin with intercorrelation of the variables and may be terminated if that is all that is needed. Insertion of sample-mean values for missing data has little effect on correlations (reducing them slightly), and may be preferable to pair-wise deletion, since the integrity of the intercorrelation pattern is maintained for multivariate analyses.

Not all researchers are aware of the severe distortions that distribution outliers can have on correlation coefficients. The effects are much more severe than with analysis of variance methods. If a case is an outlier (extremely high or low) on only one of the two variables being correlated, the effect is minimal; but if the case is an outlier on <u>both</u> vari-

bles, the effect can be drastic. The correlation can be severely inflated or deflated even in relatively large samples. Results of multivariate methods based on correlations can therefore be distorted as well, inordinately influenced by such cases. Researchers are encouraged to use Program UNITAB and/or DUOTAB to detect the presence of these cases with newly developed data sets. Deletion of such data can usually be justified in terms of non-representativeness of the population sampled.

Program MISCOR

In many situations the researcher is interested in correlations among all pairings of a set of variables, but as separate relationships, rather than as a strictly consistent pattern. This program applies the missing data option to each pairing of variables independently, thus maximixing the N on which each coefficient is based.

The program permits punched output of the correlation matrix for entry to multivariate analytic programs such as REGRAN and FACTOR, which employ row-wise deletion if used with raw data. If the amount of missing data is relatively small, this is probably reasonably safe; little distortion of the intercorrelation pattern is likely.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards

Parameter Card Fields

1-5 = number of variables (max = 100)

6-10 = number of subjects

15 = missing data option

20 = output option for correlation matrix

Input

Only score fields should be included in the format.

Output

Means and sigmas and the number of valid cases for each variable considered separately are listed first. A square matrix is then provided in which correlations appear above the main diagonal (which contains "1.0" values) and the N on which each is based appears in a corresponding position (reversed subscripts) below the main diagonal.

If punched output is produced, only the upper triangular values, including the main diagonal, will be punched -- as a continuous list. Format for these cards is (8X, 9F8.5). A total of K(K + 1)/2 values will be punched.

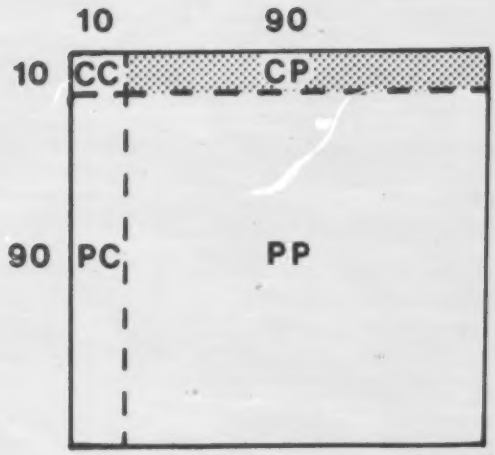
Buffers

INPUT, OUTPUT, PUNCH

RFL = 108000 with a TAPE added

Program SELCOR

Researchers often find the need to correlate a small set of criterion variables with a much larger set of predictors. Intercorrelations of each set are not of interest. The MISCOR program will handle up to 100 variables, but if it is used for a problem with 10 criteria and 90 predictors, most of the voluminous output will be unnecessary, as illustrated below.



Only the CP submatrix (or PC, its reflection) is of interest. The SELCOR program permits the user to compute and print only the particular cross-correlation submatrix needed. As in Program MISCOR, pair-wise deletion of missing data is available at option.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards

Parameter Card Fields

1-5 = number of variables in set A (max. = 25)

6-10 = number of variables in set B (max. = 100)

11-15 = number of subjects

20 = missing data option

25 = output option for means and sigmas (print only)

Input

Set A variables, followed by set B variables are read as a single list for each subject. This is easily arranged if the sets of variables are punched in separate cards. When this is not the case, an override

version of READX may be used to reorder the input vector so that the criteria are the first variables in the list.

Only score fields may appear in the format.

Output

A series of A by B matrices are printed by the program. Means and sigmas for each combination of A and B variables are optional. Following will be a matrix containing the valid N for each correlation, and then the matrix (R) of cross-correlations.

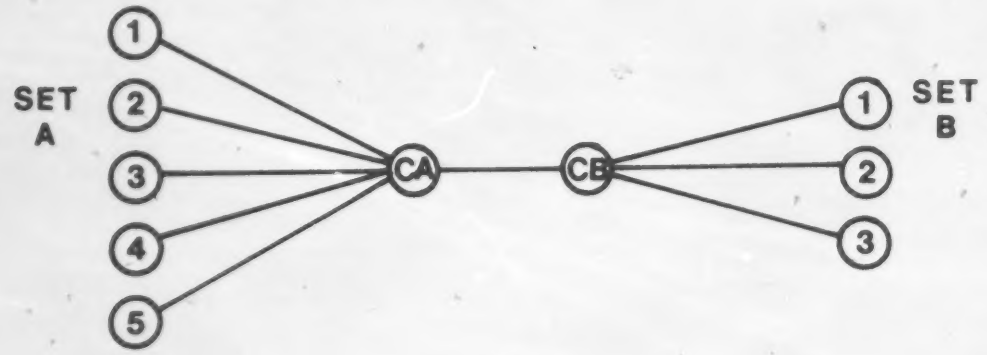
Buffers

INPUT, OUTPUT

RFL = 102000 with a TAPE added

Program CANONR

Canonical correlation is a procedure for establishing multivariate relationships between two sets of variables, referred to here as "set A" and "set B." When there is only one variable in set A or in set B, the method reduces to multiple correlation. With only one variable in each set, of course, we have a simple Pearson correlation.



In the example diagrammed above, set A consists of five variables, and set B contains three. The lines represent relationships. On each side a set of weights is defined for the variables, such that when they are used

to define the composite variables CA and CB, the two composites will be as highly correlated as possible.

Successive independent (uncorrelated) composites may be extracted up to the smaller number of variables (three in the example). Each of these canonical relationships is expressed as a coefficient "root" analogous to a squared multiple correlation, and can be tested for significance.

The composites are defined or extracted so that each successive one accounts for as much variance as possible, very much like principal axis factors. Recently, Cliff and Krus (1976) have recommended varimax rotation of the canonical structure to obtain dimensions of greater interpretability. The CANONR program provides both the unrotated and rotated solutions. Formulas for the procedure may be found in Cooley and Lohnes (1971).

Although interpretation of the canonical functions can sometimes yield useful insights, the significance test can be viewed conservatively as simply justification for examining the strongest bivariate relationships in the cross-correlation matrix.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards

Parameter Card Fields

1-5 = number of variables in set A (max. = 50)

6-10 = number of variables in set B (max. = 50)

11-15 = number of subjects

16 = missing data option

17 = output option for correlation matrix

18 = output option for weights

19 = 1 to suppress roots with p > .10

20 = number of dimensions to rotate

Input

The format must contain only score field designations. For each subject the format describes a list of A variables followed by B variables. If the number of variables in the two sets is unequal, the larger set must be set A. Subroutine READX can be used to reorder the data if necessary.

With large numbers of variables in both sets, it is desirable to suppress output of non-significant roots, using the option in Parameter column 19. At least two functions will be output in any case, however. If column 20 is left blank or zero, the number of significant functions ($p < .10$), or two in any case, will be rotated.

If the number of subjects is entered as a negative value, a correlation matrix is expected instead of raw data; see Program MISCOR for the format of this matrix. If the missing data option is used, the sample will be automatically reduced to include only subjects with valid data on all variables.

No linear dependencies (see page 57) are permitted in either set of variables.

Output

Means and sigmas of all variables are printed, followed (optionally) by the intercorrelation supermatrix of all A + B variables.

The roots (squared canonical correlations appear next, followed by their chi-square values, degrees of freedom, and chance probabilities.

The weights for the canonical functions can be printed next, but they are not normally of interest, except to be punched for use with Program SCORES to obtain canonical scores for individual subjects.

Correlations (loadings) of the A variables with the canonical variates (columns) appear next, followed by a comparable table for the B variables.

The last part of the output is the loadings of the two sets of variables after the canonical dimensions have been rotated by the varimax method (labeled VA LOAD and VB LOAD).

Buffers

INPUT, OUTPUT, PUNCH

RFL = 61000 with a TAPE added

Regression Analysis

Basic to all forms of regression analysis is this general form of the regression equation:

$$Y = a_o U + b_1 X_1 + \ldots + b_K X_K + E$$

In this equation capital letters represent variables and lower case letters represent constants, which are weights for the scores of particular subjects. Y is the criterion variable, and the X variables are called predictors. U is a dummy variable, a value of one for every subject, making a an additive constant. The b weights may be in raw-score form, or in standardized form (applicable to z-scaled Xs). Standardized weights are called "betas"; in such an equation the regression constant a is zero.

The E variable is the error of prediction, the difference between the actual and predicted Y values. The E values are sometimes called "residual scores."

$$E = Y - \hat{Y}$$

Once the weights have been established with a sample of subjects, predicted scores may be computed for any subject, or for any hypothetical subject with a particular score pattern on the X measures.

Types of Variables

Three special kinds of variables are frequently used in regression analysis: binary variables, product variables, and powered variables.

Binary variables. A nominal variable is represented in a regression equation by a set of binary (1, 0) variables, one for each category or group. On each, a subject is coded one if he is a member of the group and zero otherwise.

Product variables. Interaction is represented in a regression equation by the addition of a variable in which each subject's score is the product of his scores on the two interacting variables. When a nominal variable is involved, the interaction is represented by a set of product variables, one for each binary.

Powered variables. Curvilinearity is represented in a regression equation by the addition of a variable in which each subject's score is a power of the predictor concerned. To allow the regression line to bend once (quadratic), the added variable would contain the square of the original score.

Comparing Equations

Research questions are answered in regression analysis by comparing the predictive efficiency of two equations, one of which contains a subset of the predictors in the other. These are sometimes called the "full" and "restricted models." The index of predictive efficiency is the squared multiple correlation (R^2). When an equation is solved with a particular

sample of subjects, a and b weights are defined so as to minimize the E values and to maximize R^2. Since the restricted model contains less potentially predictive information, its value must be equal to or less than the full model. An F ratio can be computed to test the significance of the obtained R^2 difference.

$$F = \frac{(R_F^2 - R_R^2)/(K_F - K_R)}{(1 - R_F^2)/(N - K_F)}$$

where K represents the number of independent predictors in an equation and N is the sample size. Terms on the right side of this ratio are its degrees of freedom.

R^2 can be interpreted directly as the proportion of criterion variance "explained" or "associated with" or "accounted for" by the predictors. An R^2 difference has a similar interpretation; it is called the "independent contribution" of the predictor(s) omitted in the restricted model. It is the information provided over and above that of the variable(s) which remain in both equations. Such variables that remain are called "covariables" or "control variables," and are said to be "held constant."

Independent Predictors

A predictor variable is said to be independent if it cannot be perfectly predicted from the other variables in the set. Because the unit vector is always implicitly present in every regression equation, one of any mutually exclusive set of binaries is redundant, since the set sums to the unit vector. In the following example, the original continuous X variable is redundant also, because the products with binaries A, B, and C sum to it. Thus, the number of independent predictors (counting U) is six.

$$Y = U + A + B + C + X + XA + XB + XC$$

There are a variety of texts on regression analysis. Ward and Jennings (1973) provide a careful introduction to the subject, and Kerlinger and Pedhazur (1973) offer a very readable and comprehensive coverage of the theory and applications.

Regression Programs

Three programs are included in this section. One could argue that Program CANONR belongs here also, but all programs considered here concern regression analysis with a single criterion and multiple predictors.

Program REGRAN is a general purpose regression routine, allowing the user to specify a series of models and comparisons of them, designating for each model any variable as the criterion and any set of others as predictors.

Program COVARY carries out a predetermined series of model reductions for a very common analysis of covariance design: two to five groups of subjects and one covariable.

Program COMMAP implements a procedure called "commonality analysis," in which the predictable variance of a criterion is partitioned into all possible independent and joint contributions associated with two to nine sets of predictors.

Some computer programs do not permit the inclusion of linear dependencies (redundant variables) in the equations. All programs in this section employ an iterative approximation method for solving the equations, and therefore are not restricted in this way.

Regression vs. Analysis of Variance

It is not always understood that any analysis of variance can be

expressed as appropriate comparisons of regression models. However, regression analysis has at least two important advantages.

Researchers often dichotomize or trichotomize scores obtained as continuous measures in order to define levels for independent variables in anova. Not only does this waste a great deal of information, it also may badly distort the way the variables are actually related in the population sampled. Regression analysis permits use of the original continuous measures as predictors, and registers precisely their independent as well as interactive or non-linear relationships with the (criterion) dependent variable.

The other major advantage of regression analysis is that it allows the researcher to test precisely the hypothesis of interest, and not simply search among the predetermined set of anova "effects" for his answer, none of which may be a direct test of the question he has posed. Such precise tests also have the advantage of maximizing degrees of freedom for the error term in a significance test.

Program REGRAN

This program allows the user to enter a list of variables which are intercorrelated as the first step in the processing. Then, with special control cards, a series of regression equations (models) are defined by designating one variable as the criterion and one or more others as predictors. Finally, another special control card may be used to specify the comparison and significance testing of particular pairs of models. It is the user's responsibility to be sure that such comparisons are legitimate, i.e. that the restricted model contains only a subset of the full model predictors.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards

(Model cards)

(Test card)

Parameter Card Fields

1-5 = number of variables (max. = 99)

6-10 = number of subjects

11-15 = number of models (max. = 50)

16-20 = number of tests (max. = 20)

21 = missing data option

22 = output option for means and sigmas

23 = output option for intercorrelations

24 = output option for beta weights

25 = output option for raw weights

Model Card Fields

1-6 = model label

7-8 = criterion variable number

9-10 = number of independent predictors

11-80 = binary predictor selection string. Continue on second card in col. 11 if necessary.

Test Card Format

A pair of two-digit model numbers designates each comparison, beginning

with col. 1-4, 5-8, etc. The full model number is designated first in each pair: To test a model against a zero null hypothesis, enter 00 as the second model number of a pair.

Input

The format must specify only score fields for the variables. If the missing-data option is used, the sample will automatically be restricted to include only subjects with complete data. If the number of subjects is entered as a negative value, the program will expect a correlation matrix to be entered in place of data cards. See Program MISCOR for the format of such a matrix.

The model card label can be any set of six characters or blanks. The number of independent predictors includes the unit vector, which is implicit in any model, but does not include any redundant variables, even if they are specified as predictors. The binary predictor selection string uses each column beginning with 11 to indicate whether a variable is (punch one) or is not (blank) a predictor in the model concerned.

Subroutine READX is especially useful with this program, since interaction product variables and curvilinear powered variables need not be punched, but can be generated when an analysis is carried out. For instance, to test whether A and B interact to affect Y, that is, whether they have a joint effect over and above additive effects, these equations need to be compared:

$$Y = A + B + AB$$

$$Y = A + B$$

The AB variable can be generated in READX, as can the X^2 variable needed to test for a curvilinear relationship between X and Y.

$$Y = X + X^2$$

$$Y = X$$

Output

The program optionally prints means, sigmas, and an intercorrelation matrix first, using row-wise deletion of missing data. If that is all the user needs, the number of models and tests may be indicated as zero. In most regression analyses means and sigmas should be printed simply as a check, but the correlation matrix is not of interest, since validity coefficients (bivariate correlations of predictors with the criterion) will be printed with the output of each model solved.

For each model the criterion variable number and the number of independent predictors (NP) is indicated and then a table of coefficients (labeled "Regression Equation") is printed. For each selected predictor variable (V) the beta weight, the raw weight, and the validity coefficient are listed. The regression (R) constant (a weight) is also reported.

For each model comparison the R^2 (RSQ) values of the full (first) and reduced (restricted) models are repeated, followed by the difference between them. Rounding error in the iterative process may make this difference slightly negative occasionally. Degrees of freedom for the numerator (DFN) and denominator (DFD), the F-ratio, and its chance probability are then reported.

If punched output of weights is selected, each model will be punched as a separate vector, with zero weights for the criterion and any unselected predictors. The regression constant will be added to the end of a raw weight vector. The format for these cards is (A8, F2, 7F10.4) with the model label in the first field and card number in the second.

Buffers

INPUT, OUTPUT, PUNCH

RFL = 54000 with a TAPE added

Program COVARY

The term "analysis of covariance" has been used by various authors to label a variety of statistical procedures. In general, the purpose of this method is to compare regression lines fitted to the subject-point swarms of two or more groups under various restrictions. Program COVARY carries out a pre-determined series of such comparisons, using increasingly restrictive models.

The five regression models computed for each analysis could be run and compared with Program REGRAN, but COVARY is much more convenient for problems of this common type.

Model One

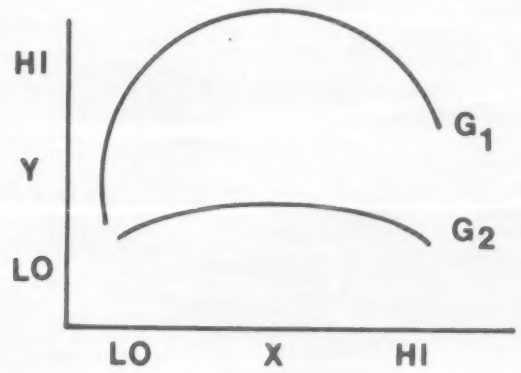
The first model computed is as follows:

$$\hat{Y} = U + X + G_1 + G_2 + G_1X + G_2X + X^2 + G_1X^2 + G_2X^2$$

This is the equation for a two group problem. With more groups there would be additional G, GX, and GX^2 vectors. The G vectors are binary group-membership variables. X is the covariable and Y is the criterion (dependent) variable. The vectors containing squared X scores permit the regression lines to take a quadratic form (one bend). The product vectors (GX and GX^2) permit the form of the regression lines to differ among the groups. The G vectors permit the regression lines to have different overall levels (intercepts) on Y.

Because $G_1 + G_2 = U$, $G_1X + G_2X = X$, and $G_1X^2 + G_2X^2 = X^2$, there are six

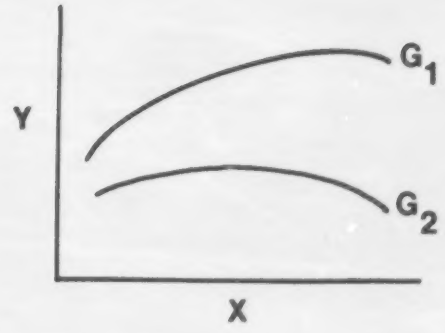
independent predictors in this first model. It is the least restrictive of those computed by the program. The following graph exemplifies a situation where all aspects of the lines for the two groups differ.



Model Two

$$\hat{Y} = U + X + G_1 + G_2 + G_1X + G_2X + X^2$$

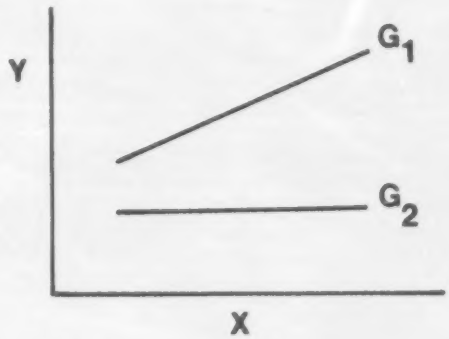
The restriction imposed in this model is that the quadratic components of the two groups must be the same. The lines may curve, but they must curve to the same degree and direction. Note that G_1X^2 and G_2X^2 have been omitted to force this restriction. If the R^2 for this model is significantly smaller than that for model one, we conclude that important predictive information is contained in those two vectors -- that the two swarms of subject-data points are not as well fitted -- that the best-fitting regression lines do have different curvatures. Note that the linear components of the lines are not involved in this comparison; the presence of G_1X and G_2X allows them to differ. The graph of the previous example would change to reflect the restriction imposed:



Model Three

$\hat{Y} = U + X + G_1 + G_2 + G_1X + G_2X$

Now we remove all squared-score vectors, forcing the regression lines to be straight. If this results in a significant loss in R^2, we conclude that the lines do indeed curve. The graph now reflects this restriction. Most of the research reports based on covariance analysis begin with this model, ignoring the possibility of curvilinearity.

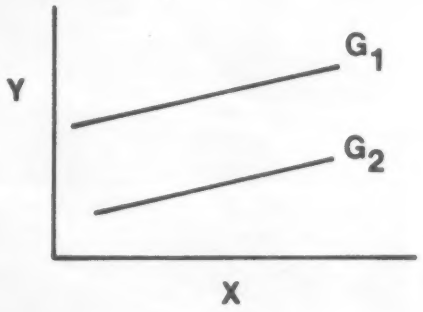


Model Four

$\hat{Y} = U + X + G_1 + G_2$

This model does not permit the two regression lines to have different slopes; they are forced to be parallel. The comparison of models three and four is therefore sometimes called the "slopes test." If it is signif-

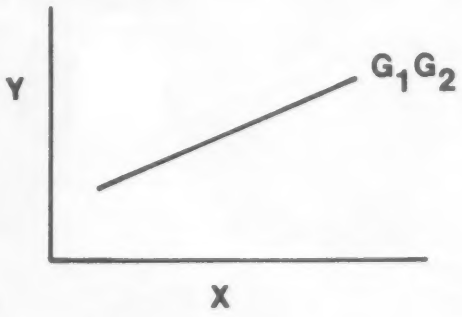
icant, we conclude that the relationship of X to Y is not the same in the two groups. Some researchers do not even make this comparison, but simply assume the lines to be parallel. This test, however, is the one of primary interest in ATI (Aptitude/Treatment Interaction) studies, particularly when the regression lines cross near the center of the graph, indicating that which is the better treatment depends on the subject's covariable (aptitude) score level.



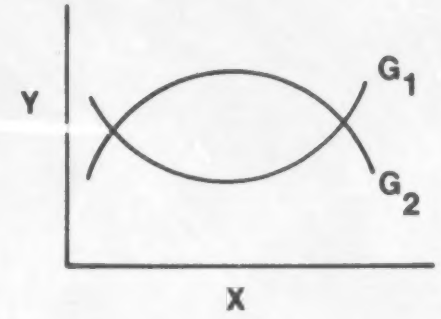
Model Five

$\hat{Y} = U + X$

In this last model the two lines are forced to merge. Comparison of this model with model four tells us the degree to which the two regression lines intercept the Y axis at different levels. If it is significant, we conclude that subjects in group one score higher on Y, no matter what lev l of the covariable is involved.



It is entirely possible for a complex component of the regression lines to be significant when nothing else is. For example, in the following graph, only the difference in quadratic components is important.



Consideration of the effects of subsequent restrictions will show why this is so. If we force common curvilinearity on the lines, the reverse curvatures cancel each other, leaving one flat straight line representing both groups.

When more than one model comparison is statistically significant, the user should plot the regression lines from the model preceding the first significant drop in R^2. Any significant effects appearing subsequently will be apparent also in the resulting graph.

Expected Criterion Scores

For each model, five scores are computed for each group of subjects. These values are the expected (predicted) scores of people at five different levels of the covariable ($\mu \pm 1$ and 2σ). The actual values of the covariable (baseline) are listed at the top of the output page. The expected scores, then, are criterion (ordinate) values. By connecting the five points for each group, the shape of the regression lines can be determined.

If only the intercepts test is significant, regression lines need

not be plotted. The only scores of interest are the expected criterion scores at the mean of the covariable; these are sometimes called "adjusted means."

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards, subdivided by Group cards

Parameter Card Fields

1-5 = number of dependent variables (max. = 100)

6-10 = number of groups of subjects (2-5)

15 = missing data option

Input

The format must contain only score-field specifications. Input order is of all covariables, followed by their corresponding criterion (dependent) variables. If the missing data option is used, any subject missing either a criterion or covariable score will be excluded from that particular analysis.

Output

The first page contains information from the initial control cards and the Group cards.

Output for each analysis appears on a separate page. The valid N for each group is given first and then the results of each of the five models along with the tests of each drop in R^2. The overall sigma of the criterion variable appears above the listing of expected scores;

the criterion mean is the middle expected value for the last model. The overall mean of the covariable is the middle value above the first set of expected scores; the corresponding sigma may be computed by subtracting the value to the left of this.

The covariable levels should be examined to be sure they are all within the actual range of the data; with badly skewed variables, the high or low points may not be of interest (or even possible).

Buffers

INPUT, OUTPUT, TAPE2

RFL = 42000 with a TAPE added

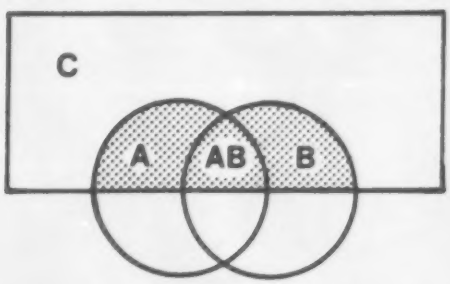
Subsidiary Routine

MODEL

Program COMMAP

When a researcher employs the multiple correlation method, he is usually interested in more than the R^2 value obtained. He would also like to determine the relative contributions of the predictors to explanation of the criterion variance. A variety of indices have been used for this purpose.

Commonality analysis completely partitions the predictable criterion variance into all possible independent and joint contributions. Consider the following Venn diagram where the rectangle represents the criterion variance and the two circles represent the variances of two predictors.



The shaded area represents the multiple correlation of A and B with C. The independent contributions, labeled A and B, can be computed by subtraction.

$$I_A = R^2_{AB} - R^2_B$$

$$I_B = R^2_{AB} - R^2_A$$

Normally, the sum of the independent contributions will be less than the total R^2. In the diagram the remaining variance is labeled AB and is called a "joint contribution."

$$J_{AB} = R^2_{AB} - I_A - I_B$$

A joint contribution registers what two or more variables have <u>in common</u> that is predictive of the criterion. These are not interaction effects. Interpretation of a joint contribution is a challenge to the researchers ingenuity (or imagination), but at least the relative contributions of all the sources are made explicit in an unambiguous metric.

Commonality analysis is not restricted to using single variables as predictors. <u>Sets</u> of variables may be included or dropped in the equations together. With sets of very different sizes and relatively small samples, however, differential degrees of R^2 inflation may make interpretation difficult.

It is possible for a joint contribution to be a negative value -- a seemingly absurd result. One cause of this is the presence of a "suppressor effect" signaled in ordinary regression analysis by opposite signs on a validity and the corresponding beta weight. Generally stated a negative joint contribution is an indication of an inconsistency at a particular point in the intercorrelation pattern. The result is that

the independent contributions of the variables concerned will be inflated to that extent.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Set assignment card

Data cards

Parameter Card Fields

1-5 = number of variables (max. = 100)

6-10 = number of subjects

11-15 = number of predictor sets (max. = 9)

16-20 = criterion variable number

21 = missing data option

22 = output option for correlation matrix

Set Assignment Card

This is a standard format option control card which designates for each variable in turn the set to which it is assigned. Unused predictors and the criterion are designated blank or zero. A "set," of course, may contain only one variable.

Input

The format must designate only score fields. If the missing data option is used, any subject with one or more missing scores will be excluded from the analysis. If the number of subjects is entered as a negative value, a correlation matrix (see Program MISCOR for the format)

will be expected in place of the data cards. In order to ensure correct internal computation of degrees of freedom, linear dependencies should be avoided in the full list of variables.

Output

After printing the assignment card contents, means and sigmas of all variables are printed, along with a list of validity coefficients. The full intercorrleation matrix may be printed at option. The total R^2 for all variables together is printed with its chance probability.

With nine sets of variables a total of 502 joint contributions can be computed. To keep the amount of output manageable, only joint contributions of more than one percent of the predictable criterion variance are printed. The rest are summed as "residual effects." All independent contributions are printed.

Each source (set combination) yields a separate line of print. The R^2 values for equations containing the variables concerned are labeled TOTAL; the corresponding independent or joint contributions are labeled UNIQUE. The percentage that each is of the R^2 for the full equation is labeled PCT. Tests of significance are computed for both TOTAL and UNIQUE for all independent contributions, but cannot be computed for the joint effects.

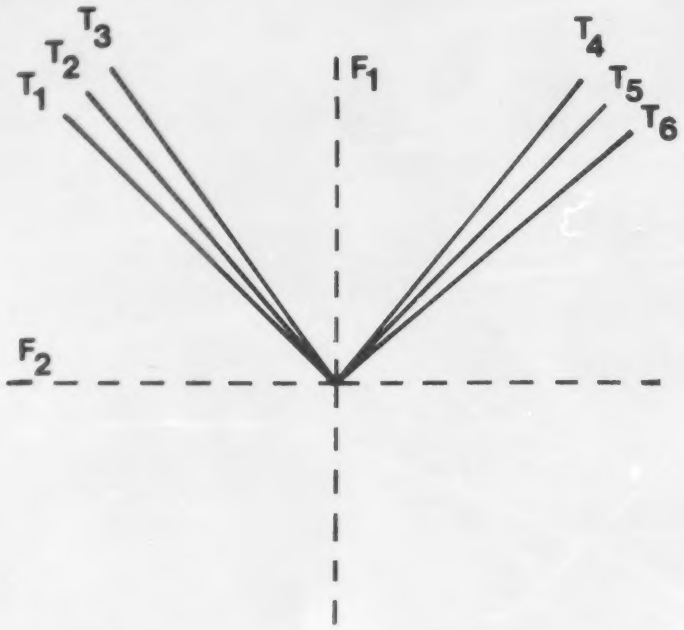
Buffers

INPUT, OUTPUT

RFL = 55000 with a TAPE added

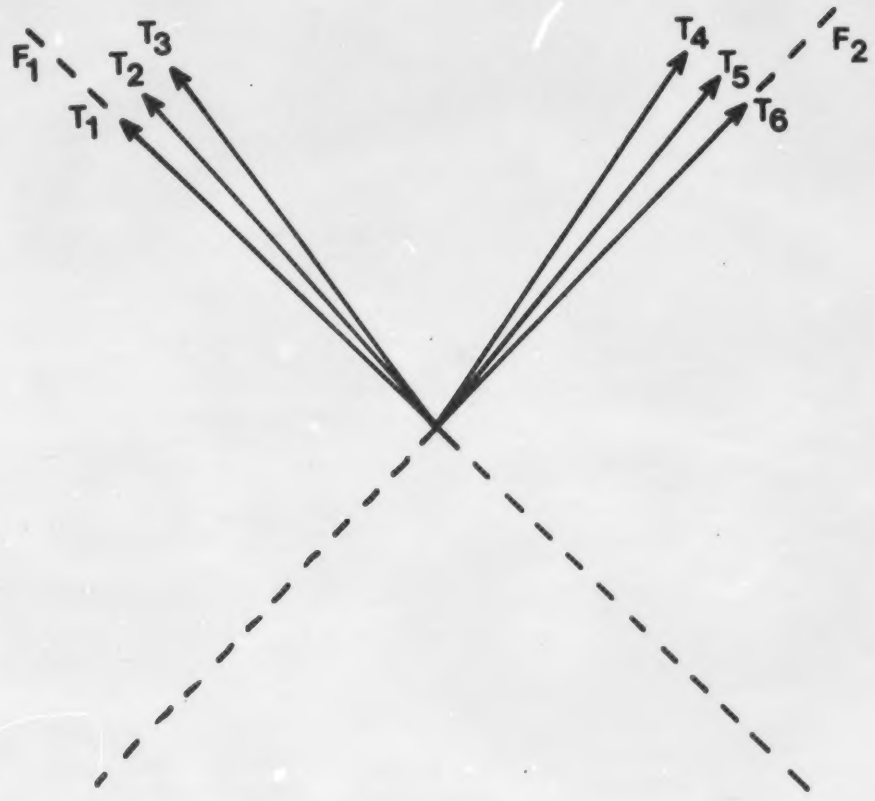
Factor Analysis

Factor analysis may be defined as a method for constructing a small number of independent (uncorrelated) composite variables (factors) from a large number of correlated source variables (tests or items). Factor variables are weighted composites of the same nature as those described in the section on Program SCORES. Correlations of factor variables with the variables used to define them are called "loadings." Factors may be considered as dimensions defining a space in which the test vectors are located. In the example below two factors define a two-dimensional space in which six tests are located.



Angles in this diagram correspond to correlations -- between tests as well as between factors and tests. In the forms of factor analysis provided by Program FACTOR, the factor variables are always uncorrelated (at right angles, orthogonal).

The positioning of the factor axes defining the space is arbitrary. That is, they can be rotated around their origin (centers) to any degree.



Usually, the most interpretable positioning of the axes is that closest to what is called "simple structure." A perfect simple structure never occurs with real data; it would be one where each test loaded perfectly on one factor or another. In the example above, the rotation is as close as possible to this ideal.

Many research problems concern similarity between factor structures. Program RELATE allows the user to rotate an empirically derived factor structure as closely as possible to another empirical structure, or toward an artificial hypothetical structure.

Factor scores positioning subjects on each factor dimension may be computed in a variety of ways. The FACTOR program allows regression weights to be punched for use with Program SCORES, yielding what are sometimes called "exact scores." A number of studies have found such scores to have inferior external validity due to "overfitting" the data on which the weights are based. Better composite measures are Likert scales based

on the items which load each factor most strongly. Such Likert scales may be moderately intercorrelated (oblique structure); in the example diagram they would better fit the clusters of tests than would exact scores.

One of the most important decisions the factor-analyst must make is how many factors to rotate. The most common error is over-factoring. Program FACTOR provides an objective criterion for this decision (Veldman, 1974).

The nature of the variables entered to a factor analysis should be carefully considered. Artifacts may be introduced by subsets of items with dissimilar formats or sets of items that are hydraulically related, such as a set of percentages.

Program FACTOR

This program implements the most widely used factor analytic procedure and one important optional variant. There are four major steps in the processing.

1. Intercorrelation of the variables
2. Conversion to image covariances (optional)
3. Extraction of principal components
4. Varimax rotation of factor structure

Image analysis involves factoring an adjusted correlation matrix (G) in which only common (shared) variation of the tests is represented. The main diagonal contains R^2 of each variable with the others, and the off-diagonal elements are complements of partial correlations. In the previous diagrams this would result in shortening the test vectors to various degrees; tests with more common variance would determine the factor

structure more than would other tests.

Principal axis factors are defined sequentially so that each accounts for as much of the remaining variance as possible. Therefore, the first principal axis empirically defines the central thrust or theme of the full set of items, and is sometimes useful for this reason. Usually, multiple-factor structures are sought, however, and the principal components structure is not of interest.

The sum of squares of a column of factor loadings is called an "eigenvalue" or "root." It is an index of the amount of variance accounted for by the factor. As many factors as there are tests can be extracted, but seldom are. The sum of the eigenvalues of all possible factors will equal the sum of the diagonal values of the basic matrix factored (R or G). A rule of thumb often used to decide on the number of factors to rotate is to use only factors with an eigenvalue greater than one. Generally, this rule defines too few factors from a small number of tests, and too many factors when the number of tests is large.

The varimax rotation process seeks to rotate the principal axes as close as possible to simple structure. This amounts to maximizing simultaneously the variances of the columns of factor loadings, and usually makes the roots more equal in size. Although the values of the roots change, their sum (the total amount of variance) remains constant. The sum of squares of a row of loadings is called "communality" and indicates the proportion of variance of a test which is registered by the factors extracted.

The varimax rotation process seeks to maximize factor-column variances simultaneously, and yields a particular index of this. At option, this value can be computed for rotations of two to ten principal axes.

When the structure is reasonably strong, the rotation with the highest varimax value can be identified as the structure with the most clarity. This procedure works only with image analysis.

Input Deck Arrangement

Title card

Parameter card

Format card

(Missing data option card)

Data cards

(Label cards)

Parameter Card Fields

1 = minus sign if Label cards are used

2-5 = number of variables (max. = 100)

6-10 = number of subjects

11-15 = number of factors signal

0 = all factors with eigenvalue > 1.0

-1 = highest varimax criterion value

+ value = extract that number of factors

16 = missing data option

17 = output option for means and sigmas

18 = output option for R matrix

19 = 1 to convert R matrix to G

20 = output option for G matrix

21 = output option for PA loadings

22 = output option for PA weights

= output option for V loadings

24 = output option for V weights

25 = output option for primary loading lists

Input

The format must include only score-field specifications. If the missing data option is used, the sample will be automatically reduced to include only subjects with valid data on all variables. If the number of subjects is entered as a negative value, a correlation matrix will be expected in place of data cards. (See Program MISCOR for the format).

If the option in Parameter column 25 is taken, a pack of label cards, one per variable, may be entered after the data deck as a supplement to the variable numbers in the printed output. All 80 columns of each card will be printed.

If the automatic varimax criterion for the number of factors is employed (Parameter col. 11-15 = -1), the image analysis option (Parameter col. 19) must be taken. Varimax weights cannot be output if this signal is used, but once the best number of factors is determined, it could be entered in Parameter col. 11-15, and the program rerun to get the V weights.

Output

If the missing data option is used N VALID will be printed for each variable, as well as the number of cases included after sample reduction. Means and sigmas are usually printed as a format check. The R and G matrices are seldom of enough interest to be printed. If the image analysis option (Parameter col. 19) is taken, the proportion of common variance for each variable and the overall percentage will be printed.

Eigenvalues will always be printed along with a commonality (h^2) estimate for each variable. Because 10 principal axes are always extracted if the varimax criterion is used to select the number of factors, these commonalities will not be relevant in such a case.

Output of principal axis loadings is normally worthwhile only if one is interested in the first principal axis. Output of varimax loadings is usually not of interest if the option (col. 25) for ordered primary loadings is taken. Output of either principal axis or varimax weights is usually only to punch them for later use with Program SCORES, which itself is seldom advisable in comparison with other methods of factor scoring.

The ordered primary loadings option results in assigning each variable to the factor it loads most highly (absolute). Then the list for each factor is ordered by the size of the loadings and printed along with contents of the label cards if Parameter column one was a minus sign. If the col. 25 option was two or three, primary loading and loading-sign lists will be punched in the form of the Scale and Reversal key cards that are used in Program SCALES, except that the reversal key will contain ones rather the number of choices for each reversed item. This punched output is limited to nine factors. Since every item will be assigned to some factor, it is usually wise to inspect the lists to eliminate items with small loadings before computing Likert scale scores for the factors.

Buffers

INPUT, OUTPUT, PUNCH

RFL = 126000 with a TAPE added

Program RELATE

The criterion for varimax rotation is internal in the sense that the simple structure of the data from a particular sample is maximized. Program RELATE allows the user to rotate any factor-loading structure -- principal axis or varimax -- toward a predetermined target matrix, which may

be another empirically-derived structure or an artificial simple structure defined a priori.

Input Deck Arrangement

Title card

Parameter card

Format card

Target matrix of loadings (or theoretical assignments)

Matrix of loadings to be rotated

Parameter Card Fields

1-5 = number of variables (max. = 100)

6-10 = number of factors in the target (max. = 25)

11-15 = number of factors to be rotated (max. = 25)

20 = output option for rotated matrix

25 = 1 if a theoretical target is to be specified

Input

If punched output of factor loadings from Program FACTOR is used as input for the second or for both matrices, the format would be (10X, 7F10.4). The format for the two matrices must be the same in any case, unless a theoretical target is specified. The number of factors to be rotated (second matrix) may not exceed the number in the target.

A theoretical simple-structure target matrix will be constructed if the option in Parameter col. 25 is used. In this case one or more cards in format (40F2) are used to enter a list of the factor numbers to which each variable is assigned. A loading matrix containing ones (plus or minus) and zeros will be constructed. Variables to be negatively signed are negatively signed on the special assignment card(s).

Output

If a theoretical target is specified, the assignments of items to factors will be printed first as a check.

Correlations of all pairings of factors in the two matrices are provided next. In this matrix rows (A) are factors in the target matrix. These are relationships between the factors before rotation toward the target.

At option the rotated second matrix (NEW B) may be output next.

The final list of coefficients printed indicates for each pair of corresponding test vectors in the target and rotated matrices how similar their placements are. These may also be interpreted as correlation coefficients. A value of .90 indicates very close correspondence of the test vectors. A coefficient of .71 signifies an angle of 45 degrees -- not very close. A zero value means 90 degrees -- no similarity at all. Negative values suggest opposite placement in the two structures, even after maximizing their overall similarity. These values are useful in pinpointing bad items when a theoretical target is used.

Buffers

INPUT, OUTPUT, PUNCH

RFL = 62000 with a TAPE added

Statistical Utility Routines

The routines in this section are employed by many different PRIME programs. Programmers constructing special-purpose statistical prodedures should find them very useful, particularly for complex matrix operations.

In the descriptions that follow, necessary information about dimensioning of arrays is provided. When scratch vectors are used as arguments, the

necessary length is indicated; when vector length or matrix columns is a function of the problem, the number one is arbitrarily indicated.

Because some of these routines themselves call other utilities, programmers should avoid all utility names for their programs or their own subroutines.

Function SUMX

This routine will return the sum or sum of squares of the elements of a vector, which may be a simple vector or one comprising a particular row or column of a matrix.

Arguments (X, KX, KN, ND)

X = array containing vector to be summed

KX = row or column number if X is a matrix

KN = number of elements in X to be summed

ND = rows dimensioned for X in the calling routine

Dimension X(ND,1)

Comments

If X is a vector, ND is irrelevant, and KX should be sent = 1.

If KX is negative, it is a row rather than a column number in X.

If KN is negative, sum of squares is computed rather than sum of scores.

Function SCAP

This routine will return the scalar (inner) product of two vectors, either or both of which may be simple vectors or particular matrix rows or columns.

Arguments (X, Y, KX, KY, KN, ND)

X = array containing one of the vectors

Y = array containing the other vector

KX = row or column number if X is a matrix

KY = row or column number if Y is a matrix

KN = number of element products to be summed

ND = rows dimensioned for X and Y in calling routine

Dimension X(ND,1), Y(ND,1)

Comments

If X is a vector, KX should be sent = 1. The same is true of Y and KY. If both are vectors, ND is irrelevant.

If KX is negative, it is a row rather than column number in X. The same is true of KY and Y.

In the calling routine, X and Y may be the same array.

Subroutine MNORM

This routine will normalize all rows or all columns of a matrix, using divisiors computed from the input matrix or received in a separate vector. For each row or column, the square root of the sum of squares is divided into each element of the vector.

Arguments (A, B, C, NR, NC, ND)

A = input matrix

B = output matrix, normalized

C = vector of divisors

NR = rows of A and B

NC = columns of A and B

ND = rows dimensioned for A and B in calling routine

Dimension A(ND,NC), B(ND,NC), C(1)

Comments

If NR is negative, rows rather than columns of A will be normalized.

If NC is negative, the divisors will be computed from A and stored in C. Otherwise, divisors are assumed to have been received in C.

A and B may be designated as the same matrix in the calling routine.

Subroutine MPROD

This routine will compute the product of two matrices, with implicit transposition of either of them.

Arguments (X, Y, Z, NR, NC, NM, ND)

X = pre-multiplier matrix

Y = post-multiplier matrix

Z = returned as matrix product

NR = rows of Z

NC = columns of Z

NM = common dimension of X and Y

ND = rows dimensioned for X, Y, and Z in calling routine

Dimension X(ND,1), Y(ND,1), Z(ND,NC)

Comments

X and Y may be the same matrix in the calling routine, but Z must be a different memory area.

If NR is negative, X is multiplied as if transposed. The same is true of NC and Y.

If NR and NC are both positive, NR is rows of X, NC is columns of Y, and NM is columns of X and rows of Y.

If X, Y, or Z is received as a vector, it will be considered to be a one-column matrix.

Subroutine MINVS

This routine computes the inverse of a nonsingular square matrix and its determinant.

Arguments (NV, R, X, Y, Z, D, ND)

NV = number of variables (rows and columns of R)

R = matrix to be inverted in place

X,Y,Z = scratch vectors

D = returned with determinant of R

ND = rows dimensioned for R in calling routine

Dimension R(ND,NV), X(NV), Y(NV), Z(NV)

Comments

If a zero determinant is detected (singular matrix), notification is printed and execution is terminated with STOP 123.

Subroutine ASCOR

This routine computes means, sigmas, and an intercorrelation matrix, using READX to acquire raw data. Automatic row-wise deletion of subjects with missing data is provided, as well as standard output options.

Arguments (NV, NS, R, A, S, V, MD, KF, KS, KR, ND)

NV = number of variables (rows and columns of R)

NS = number of subjects

R = returned with intercorrelation matrix

A = returned with variable means

S = returned with variable sigmas

V = returned with valid N for each variable

MD = received as list of missing data options

KF = received as data format (10A8)

KS = standard output option for A and S

KR = standard output option for R

ND = rows dimensioned for R in calling routine

Dimension R(ND,NV), A(NV), S(NV), KF(10), MD(NV), V(NV)

Comments

The data format in KF must include only score fields if the standard READX is used.

Only subjects with valid scores on all variables will be used, and NS will be reduced and printed if necessary. Therefore, NS should not be a constant in the CALL statement.

If R is punched, see Program MISCOR or subroutine READT for the format used.

Subroutine RTOGM

This routine is used only by Program FACTOR. It converts an intercorrelation (R) matrix to an image covariance (G) matrix and provides a standard output option.

Arguments (NV, R, W, X, Y, Z, KP, ND)

NV = number of variables (rows and columns of R)

R = received as R, returned as G

W = returned as factor-weight adjustment matrix

X,Y,Z = scratch vectors

KP = standard output option for G matrix

ND = number of rows dimensioned for R and W in calling routine

Dimension R(ND,NV), W(ND,NV), X(NV), Y(NV), Z(NV)

Comments

Since the conversion process involves computation of the inverse of R, it must not be singular.

The percentage of common variance in G is always printed. If G is not printed, its diagonal elements (estimates of common variance for each variable) will be printed.

If G is punched, the format will be the same as that of a punched R matrix from ASCOR.

Subroutine ROOTS

This subroutine computes the eigenroots and vectors of a symmetric matrix. Vectors are returned as loadings (denormalized). Three options for the number of roots to be extracted are provided, along with a standard output option.

Arguments (NV, NF, R, F, E, X, KP, ND)

NV = number of variables (rows and columns of R, rows of F)

NF = received as the number of factors criterion, returned as the number of factors extracted

R = square symmetric matrix to be factored

F = returned with factor loadings

E = returned with eigenvalues (roots)

X = row commonalities (h^2) of F

KP = standard output option for F

ND = rows dimensioned for R and F in calling routine

Dimension R(ND,NV), F(ND,NV), E(NV), X(NV)

Comments

If NF = 0, only factors with roots exceeding 1.0 will be extracted.

If NF is greated than zero, that many factors will be extracted.

If NF is less than zero, a maximum of ten factors will be extracted.

In all cases the maximum number of possible factors is NV. Since NF is returned as the number of factors extracted, it should not be a constant in the CALL statement.

The R matrix is destroyed by this routine.

Eigenvalues (roots) (column sums of squares of F) are always printed. Row sums of squares (communalities) will be listed if F is printed.

Subroutine ROOTA

This routine is similar to ROOTS, except that it is intended to extract factors from a square but non-symmetric matrix.

Arguments (NV, NF, R, F, E, X, Y, Z, ND)

NV = number of variables (rows and columns of R, rows of F)

NF = number of factors to be extracted (columns of F)

R = asymmetric square matrix to be factored

F = returned with factor loadings

E = returned with eigenvalues (roots)

X,Y,Z = scratch vectors

ND = rows dimensioned for R and F in calling routine

Dimension R(ND,NV), F(ND,NV), E(NV), X(NV), Y(NV), Z(NV)

Comments

The R matrix is destroyed by this routine.

Subroutine VARRO

This routine rotates an orthogonal factor structure by the normalized varimax method. The varimax criterion value (index of simple structure) is returned, and a standard output option is provided.

Arguments (NV, NF, V, X, Y, Z, VC, KP, ND)

NV = number of variables (rows of V)

NF = number of factors (columns of V)

V = factor loading matrix to be rotated

X = returned with column sums of squares

Y = returned with row sums of squares

Z = scratch vector

VC = varimax criterion value attained

KP = standard output option for V

ND = rows dimensioned for V in calling routine

Dimension V(ND,NF), X(NV), Y(NV), Z(NV)

Comments

Row and column sums of squares will be printed if V is printed.

Subroutine SOLVE

This routine employs an iterative procedure to solve a multiple regression equation, given a matrix of intercorrelations. Any variable may be designated the criterion, and any subset of variables may be designated as predictors.

Arguments (KC, NV, LP, R, B, W, R2, ND)

KC = criterion variable number

NV = number of variables (rows and columns of R)

LP = predictor selection list

R = intercorrelation matrix

B = returned with beta weights

W = scratch vector

R2 = returned as the squared multiple correlation

ND = rows dimensioned for R in calling routine

Dimension LP(NV), R(ND,NV), B(NV), W(NV)

Comments

The R matrix may be singular, since an iterative procedure is used, rather than the inverse of R.

The list LP contains ones for selected predictor variables and zeros for all others, including the criterion.

A full list of NV weights will be returned in B; these will be zero for the criterion and unselected predictors.

Function PROB

This routine returns a chance probability value, given an F ratio and its degrees of freedom.

Arguments (DN, DD, F)

DN = numerator degrees of freedom

DD = denominator degrees of freedom

F = F ratio to be evaluated

Comments

Statistics other than F may be evaluated also:

$F = t^2$ with df = 1 and (N-2)

$F = \chi^2$ with df = df and 100000

$F = z^2$ with df = 1 and 100000

Input/Output Utility Routines

The routines described in this section simplify input and output of data in standard formats, as well as particular data manipulation processes common to many of the PRIME programs.

Programmers wishing to construct general purpose programs to, in effect, extend the capabilities of the PRIME system will find many of these routines useful.

Subroutine READC

This routine is used by all PRIME programs except CHICHI to read and print the contents of the three initial control cards: Title, Parameter, and Format.

Arguments (KA, KB, KC, KD, KE, KF, KP)

KA-KE = five-digit fields from col. 1-25 of the Paramter card

KF = contents (10A8) of the Format card

KP = program name (A6) for the initial print line

Dimension KF(10)

Comments

If a blank title card is read, KA will be returned with a zero value and nothing else will be read or printed.

Subroutine READM

This subroutine reads one or more standard (80I1) option cards, or sets an option list equal to a particular value. The option list may also be printed.

Arguments (KM, NV, MD, LL)

KM = option type

NV = number of signals

MD = option signal list

LL = label to be printed (max. = A8)

Dimension MD(NV)

Comments

If KM = 0, 1, or 2, all signals will be returned equal to KM. If KM = 3, one or more option signal cards will be read into MD.

If LL is not received blank, the signal list will be printed with this label.

Subroutine READT

This routine reads and reflects a list of values defining the upper triangular portion of a symmetric matrix, such as those punched by Programs MISCOR and FACTOR for intercorrelations.

Arguments (R, NV, KF, KP, ND)

R = returned as a square symmetric matrix

NV = number of variables (rows and columns of R)

KF = received as the data format (10A8)

KP = standard output option for R

ND = rows dimensioned for R in the calling routine

Dimension KF(10), R(ND,NV)

Comments

If the input deck was punched by a PRIME program (Subroutine ARRAY), the data format is (8X, 9F8.5).

The upper triangular matrix, including the diagonal elements, is expected as a continuous list. This is the order for a four-variable example:

1	2	3	4
	5	6	7
		8	9
			10

The contents of the lower triangle will be filled by reflecting values above the diagonal before printing and return.

Subroutine READX

The standard version of the routine is described here. See the section on Data Modification and Generation for examples of applications in which a replacement version of READX is employed. This routine is called repeatedly in most PRIME routines to acquire data from each source (usually subject) in turn.

Arguments (ID, X, KC, KF, NV)

ID = subject identification (max. = A8)

X = returned as a vector of scores

KC = received as a serial call number

KF = received as the data format (10A8)

NV = number of scores to be returned in X

Dimension KF(10), X(NV)

Comments

If a subject identification block is to be returned, ID should be set = 12345 on each CALL. Since the A field in the format will be used to return subject identification, the 12345 should be a variable in the CALL, not a constant.

Subroutine SPLIT

The primary purpose of this routine is to follow a call of READC, when a five-digit field on a Parameter card actually represents five one-digit specifications, which need to be separated.

Arguments (KX, KA, KB, KC, KD, KE)

KX = received as a five-digit number

KA-KE = returned with the five separate digits

Function MDAT

The purpose of this function is to determine, given a particular missing-data option signal, whether a particular score is to be considered valid or not.

Arguments (KO, X)

KO = missing data option signal

X = score to be evaluated

Comments

If KO = zero, X is always valid.

If KO = one, both blank and zero X are not valid.

If KO = two, only blank X is not valid.

A minus zero score is generated by a blank field.

MDAT is returned = 1 if X is valid, and = 0 if it is not.

Subroutine ARRAY

This is a general-purpose routine for printing or punching of arrays with standard formats, spacing and labeling.

Arguments (X, NR, NC, KL, KO, ND)

X = array to be output

NR = number of rows of X to be output

NC = number of columns of X to be output

KL = label for the output

KO = standard output option

ND = number of rows dimensioned for X in calling routine

Dimension X(ND,NC)

Comments

If X is a vector, its length is NR, NC is set = 1, and ND is irrelevant.

The length of the KL label depends on the type of output chosen. Print format is (A8, 10F11.4). Punch format for a matrix is (A6, 2I2, 7F10.4); for a vector it is (A8, I2, 7F10.4).

If KO is received as a negative value, X is assumed to be a square symmetric matrix, and only the upper triangular matrix will be punched, including the main diagonal elements, as a continuous vector in format (A5, I3, 9F8.5). See Subroutine READT for the order.

Subroutine SCORP

This routine punches an identification block and a vector of scores for one subject in one of four standard formats.

Arguments (ID, NV, X, KO)

ID = subject identification block

NV = number of scores to be punched

X = vector containing scores

KO = output format option

Dimension X(NV)

Comments

Options for KO are as follows:

1 = (A8, I2, 14F5.2)

2 = (A8, I2, 14F5.0)

3 = (A8, I2, 7F10.4)

In all cases, I2 holds the card number.

If KO is received as a negative value, the format (A5, 2I2, 1X, 14F5.2) will be used, and the absolute value of KO will be placed in the first I2 field.

Subroutine SWAP

This routine allows trading the contents of two memory locations in one statement instead of three.

Arguments (A, B)

A and B values are swapped.

Subroutine ALTER

This is a dummy routine used only by Program UNITAB. It is called once for each subject between the accumulation of statistics and the compilation of two-digit integer frequencies. A replacement version allows modification of scores between the two stages of processing.

Arguments (X, NV)

X = score vector

NV = number of scores

Dimension X(NV)

Subroutine MCARD

This routine is used only by Program REGRAN to read information from a Model card. It may be replaced with a user version in situations where model definitions can be systematically generated, and punching of many model cards obviated.

Arguments (M, ML, KC, NP, KP, NV)

M = serial model number

ML = model label (A6)

KC = criterion variable number

NP = number of predictors

KP = list of predictor selection signals (1,0)

NV = total number of variables

Dimension KP(NV)

Comments

Only M and NV are received from the calling routine. All other arguments are defined from the model card which is read, or by other means in a user replacement version.

See Program REGRAN for further discussion of Model card contents.

Subroutine SORT

This routine is not used by any of the PRIME programs, but is included in the system because many applications of the system, especially with data on tape or permanent files, require presorting of a data file prior to calling a PRIME program. This routine employs the UT Computation Center routine SRTMRG, but simplifies its use.

Arguments (KI, KO, KD, KR)

KI = input unit number (1-6)

KO = output unit number (1-6)

KD = number of data elements per record

KR = 1 to remove key, 0 = leave it on sorted records

Comments

KI and KO may be the same unit.

Each record is assumed to hold a single key value, followed by KD data elements. Records will be sorted from low to high key values. The key and data may be either integer or real mode.

Both KI and KO are rewound before and after sorting.

The SRTMRG routine requires a minimum field length of 77000 locations. To insure this, whether or not data are to be read from tape, the following control card sequence should be used instead of the usual EXECPF card.

```
RUN, S.
READPF, 7037, PRIMLIB.
(READPF if permanent file used)
RFL, 77000.
NOREDUCE.
(REQUEST if tape used)
LOAD, LGO, PRIMLIB.
EXECUTE.
```

With many PRIME programs the RFL value will have to be greater than 77000.

Acquisition of Source Code

Individuals or organizations outside the University of Texas at Austin who wish to obtain the source code (Fortran) for the PRIME system should write to the following firm for information and current prices.

Data Science Associates
Box 71
Austin, Texas 78767

References

Cliff, N. and Krus, D.J. Interpretation of canonical analysis: rotated vs. unrotated solutions. Psychometrika, 1976, 41, 35-42.

Cooley, W.W. and Lohnes, P.R. Multivariate Data Analysis. New York: Wiley, 1971.

Cronbach, L.J. Coefficient alpha and the internal structure of tests. Psychometrika, 1951, 16, 297-334.

Ebel, R.L. Estimation of the reliability of ratings. Psychometrika, 1951, 16, 407-424.

Kaufman, M.J.; Semmel, M.I.; and Agard, J.A. Mainstreaming: Learners and Their Environments. Baltimore: University Park Press, in press.

Kerlinger, F.N. and Pedhazur, E.J. Multiple Regression in Behavioral Research. New York: Holt, Rinehart, and Winston, 1973.

Kirk, R.E. Experimental Design: Procedures for the Behavioral Sciences. Belmont, CA: Brooks/Cole, 1968.

Rasch, G. Probabilistic models for some intelligence and attainment tests. Denmark: Denmarks Paedogogiske Institut, 1960.

Veldman, D.J. Fortran Programming for the Behavioral Sciences. New York: Holt, Rinehart, and Winston, 1967.

Veldman, D.J. EDSTAT-V User Manual. Austin: ibid, inc., 1971.

Veldman, D.J. VSTAT User Manual, Austin: ibid, inc., 1974.

Veldman, D.J. Simple structure and the number of factors problem. Multivariate Behavioral Research, 1974, 9, 191-200.

Ward, J.H. Hierarchical grouping to optimize an objective function. American Statistical Association Journal, 1963, 58, 236-244.

Ward, J.H. and Jennings, E. Introduction to Linear Models. Englewood Cliffs, N.J.: Prentice-Hall, 1973.

Wright, B.D. Solving measurement problems with the Rasch model. Journal of Educational Measurement, 1977, 14, No. 2, summer.